DMA

N° 15

DOCUMENT TECHNIQUE D'INFORMATIONS

SYSTEMES INDUSTRIELS

GROUPE BFM

METROLOGIE

Trois types d'informations sont contenus dans ce DMA N°15 :

- Vous allez découvrir les nouvelles fonctionnalités offertent aux utilisateurs **du réseau NDS-II** ainsi que celles proposées pour **l'outil de développement iMDX SERIE-IV.** C'est par des extensions de Logiciel que ces deux familles de produits gagnent en performance et s'ouvrent de plus en plus à la conduite de projet Informatique prétentieux.
Le NRM est maintenant doté du même interface humain que le NDX SERIE-IV (touches fonctions programmables, complémentation de la syntaxe, etc...). Le nombre d'utilisateurs connectés sur le réseau NRM est étendu à 30. La vitesse de compilation du iMDX SERIE-IV est augmentée dans un ratio de 5 à 10 %.....

Je vous laisse découvrir le.... RESTE.

- Toujours avec le soucis d'apporter une solution globale, METROLOGIE s'est entourée de constructeurs qui enrichissent le catalogue de base iNTEL. Dans ce cadre nous sommes distributeur notamment de **PLESSEY, T.I.T.N., QUANTUM, ...C'est un bon choix !** Tous ces constructeurs continuent a investir avec l'objectif de proposer de nouveaux produits qui correspondent aux besoins du marché. c'est ainsi que vous allez trouver des cartes au format SBC tant attendues (répeteur de MULTIBUS, entrée-sortie série 8 voies, etc...) et **un Logiciel X25** homologué par tous les réseaux publics (TRANSPAC notamment).

- La tendance est de concevoir des systèmes informatisés de plus en plus décentralisés. D'où l'existence des réseaux locaux. Dans le cadre des applications Industrielles iNTEL propose un nouveau réseau local utilisant un support de communication parfaitement défini et nommé le **BIT BUS.** Ce réseau local assure des échanges de ressources entre cartes d'entrée-sortie supervisés par une machine iRMX 86. Par ailleurs METROLOGIE distribue le réseau local **GIXINET de la Société GIXI.** Ce réseau, déjà très largement répandu (plus de 300 noeuds sont installés) assure une interconnexion entre machines organisées autour du MULTIBUS avec des débits d'échanges de 1 M bits/s. Dans un environnement de type Industriel.

..... BONNE LECTURE DE CE DMA .....

Claude GERMAIN
Directeur Technique
Division Systèmes Industriels

S O M M A I R E

****

- 1 - FAISONS LE POINT

- 2 - NDS-2 (NOUVELLE RELEASE)

- 3 - iMDX SERIE-4 (NOUVELLE RELEASE)

- 4 - LE BIT BUS

- 5 - LOGICIEL X25

- 6 - CARTES COMPATIBLES DE TITN

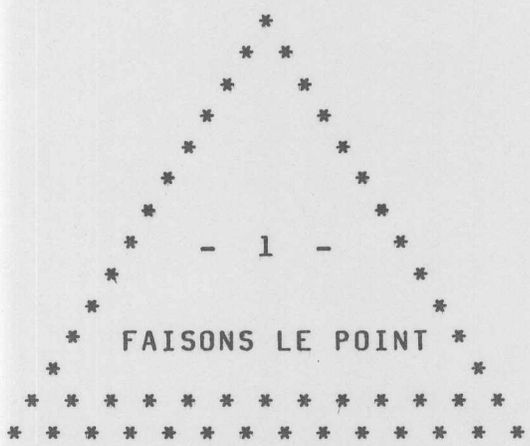
- 7 - CARTES COMPATIBLES PLESSEY

- 8 - MODULE DE QUANTUM

- 9 - GIXINET

S O M M A I R E

****


- 1 -
FAISONS LE POINT

- 1 -
FAISONS LE POINT

SWINDON SOFTWARE MANUFACTURING : Current Available Software List.

PRODUCT (CODE)	VERSION
Series-II Based Products	
ISIS-II	V4.3
ISIS-II (W)	V4.3
ASSEMBLERS:	
ASM 48 (MDS-D48)	V4.2
*ASM 51 (MCI-51-ASM)	V2.1
ASM 80/85	V4.1
ASM 86/87/88 (MDS-387)	V3.0
ASM 89 (MDS-312)	V2.0
8086/8088 SOFTWARE DEVELOPMENT PACKAGE MDS-311 (PHASE 4.0):	
PL/M 86	V2.1
ASM 86	V2.1
LINK 86	V1.3
LOC 86	V1.3
LIB 86	V1.2
OH86	V1.2
MAINFRAME LINK (MDS-384)	V1.1
HIGH LEVEL LANGUAGES:	
FORTRAN 80 (MDS-301)	V2.2
PL/M51 (iMDX-352)	V1.1*
PL/M80 (MDS-PLM)	V4.0
BASIC 80 (MDS-320)	V1.1
CREDIT (MDS-360)	V2.1
PASCAL 80 (MDS-381)	V2.1
COBOL (MDS-380)	V1.0
TOOLBOX (MDS-363)	V2.0
8087 SUPPORT S/W (MDS-319)	V1.3
PROM PROGRAMMING SOFTWARE:	
UPM	V3.2
IPPS	V1.0
2920 SUPPORT PACKAGES:	
MCI-20-SPS	V2.0
MCI-20-SPAS	V2.0

PRODUCT (CODE)	VERSION
NETWORK DEVELOPMENT SOFTWARE:	
NDS-I	V1.1
NDS-II	V2.5*
iMDX 337	V1.0
VAX CROSS SOFTWARE:	
iMDX 341VX (ASM 86)	V2.0
iMDX 343VX (PLM 86)	V2.3
iMDX 344VX (PASCAL 86)	V2.0
IMDX 394 ASC COM LINK	V2.0
IMDX 395 ASC COM LINK	V2.0
Series-III Based Products:	
PASCAL 86 (MDS-314)	V3.0
FORTRAN 86 (MDS-315)	V2.2
PL/M 86 (MDS-313)	V2.3
MCS-96 (iMDX 355)	V1.0
ASM 86)	V2.0
LINK 86)	V2.0
LOC 86) S.III	V2.0
LIB 86)	V2.0
OH 86)	V1.0
RUN)	V2.0
IMDX 333 (PSCOPE)	V1.1
IMDX 334 (AEDIT-86)	V1.0
iMDX 335 (AEDIT-80)	V1.0
IMDX 332 (PMTS)	V2.0
IMDX 364	V1.0
IMDX 356	V1.0
286 SOFTWARE PRODUCTS:	
iMDX 321	V2.0
iMDX 322	V1.0
iMDX 323	V2.3
iMDX 324	V3.0

OMS PRODUCTS

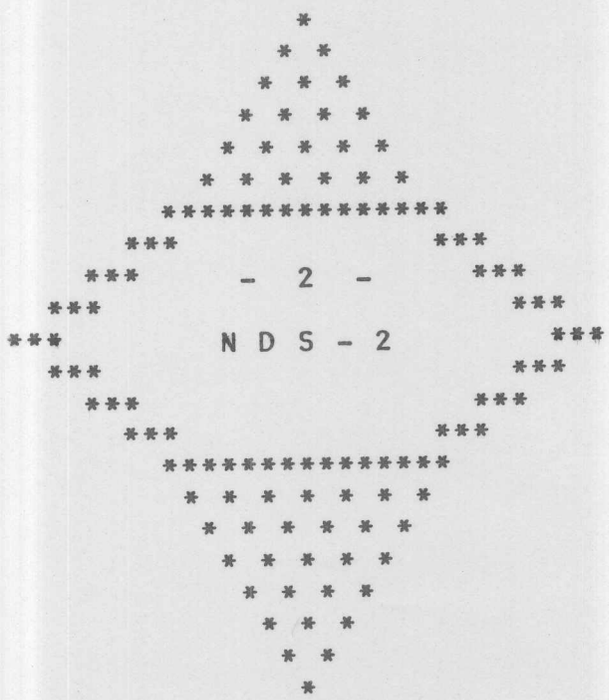
iMMX 800	V3.0	iRMX 860	V3.0
iRMX 80	V2.0	iRMX 861	V3.0
iRMX 86	V5.0	iRMX 862	V2.1
iRMX 88	V3.0	iRMX 863	V2.3*
OPL 432	V1.2	iRMX 864	V1.4
INA 960	V1.0		

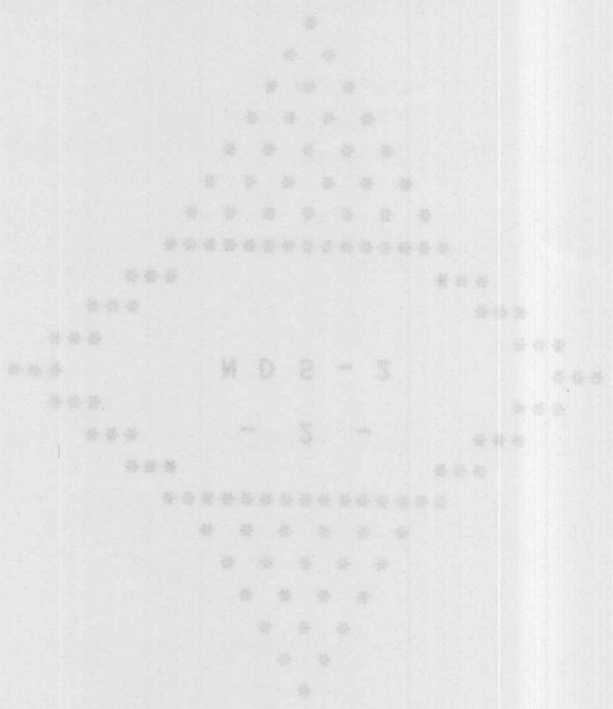
REFERENCE ONLY:

EMULATORS	
ICE 41A	V1.1
ICE 48	V1.6
ICE 49	V1.1
ICE 51	V1.3
ICE 80	V4.0
ICE 85B	V2.5
ICE 86A	V2.2
ICE 88A	V2.2
RBF-89	V1.0

* New Product or Version.

9 APRIL 1984


- 2 -
NDS-2



Le réseau NDS-2 constitué par l'association sous forme distribuée de differents outils de développement, le tout supervisé par le NRM, offre bien des avantages tant au niveau coût (lorsqu'il s'agit de créer de nouveaux postes de travail), qu'au niveau productivité (grâce au management de projet automatisé, et la distribution de travaux dans le réseau).

Une nouvelle version de Logiciel iNDX version 2.5 est maintenant disponible pour ce réseau, tandis qu'une version 2.8 sortira en Août 84. Par ailleurs une fonction "courrier électronique" sera disponible en juin 84 et il sera possible de connecter des systèmes organisés sous iRMX 86 au réseau en juin 84.

RELEASE 2.5

WHY 2.5 ?

SOFTWARE :

* RELEASE 2.5 WAS A RESULT OF INPUTS FROM THE FIELD DEMANDING THAT iNDX CUSPS AND CONVENTIONS BE UPWARDS COMPATIBLE WITH THE ISIS.

  EXEMPLE :

  SUBMIT FILE GENERATED BY THE RMX/ICU PRODUCT WOULD RUN WITHOUT PROBLEMS ON A SERIES-III BUT WOULD NOT EXECUTE ON THE SERIES-IV. RELEASE 2.5 SOLVES THE INCOMPATIBILITY ISSUE.

* PUT ADVANCED HI ON NRM FOR SUPERUSER

* PROVIDES ACL ON THE SERIES-IV

* ALLOW REMOTE OPERATION/DIAGNOSTICS OF NRM/SIV

* NEW CUSPS PROVIDED

* BUG FIX

HARDWARE :

* US AND EUROPEAN CUSTOMERS HAVE THE OPTION OF INSTALLING QUIETER FANS FOR THE SERIES-IV.

2.5 SOFTWARE WILL BE SHIPPED TO ALL REGISTERED CUSTOMERS FREE.

RELEASE 2.5

NEW CUSPS

NEW CUSPS PROVIDED ARE :

SERIES-IV AND NRM :

* STTY - SERIAL PORT CONFIGURATION PROGRAM
* SEARCH - DESIGNATE A SEARCH PATH FOR ALL CUSPS

NRM :

* NETSTAT - SHOWS LOG STATE OF EACH WORKSTATION
* NETDOWN - DISABLE LOGON TO NDS-II
* NETUP - ENABLE LOGON TO NDS-II
* NETMSG - SEND MESSAGE TO ALL SERIES-IV USERS

ISIS-III (N) :

* NETMAP - LOGICALLY ASSIGN UP TO 9 REMOTE DEVICES REGARDLESS OF PHYSICAL LOCAL STORAGE CONFIGURATION

RELEASE 2.5

NEW CUSPS

STTY : INDX CUSP THAT OPERATES ON SERIES-IV OR NRM

* THREE CONFIGURATION MODES AVAILABLE :
  - PROGRAM BAUD RATE FOR SERIAL CHANNEL 1
  - ENABLE/DISABLE TYPEAHEAD
  - REMOTE TERMINAL ACCESS TO SERIAL CHANNEL 1

    STTY BAUDRATE (1200) REMOTE

    STTY LOCAL

* TERMINAL INTERFACE CONFIGURATION FILE SIMILAR TO AEDIT MACRO FILE

* ALLOWS FOR KEYMAPPING OF THE SERIES-IV OR NRM CONSOLE. THE DEFAULT CONFIGURATION FILE SUPPLIED MAPS THE PAGE NEW/LINE KEY TO RUBOUT.

RELEASE 2.5

NEW CUSPS

SEARCH : iNDX CUSP THAT OPERATES ON SERIES-IV OR NRM

* ALLLOWS AN INITIAL SEARCH PATH FOR SYSTEM CUSPS.
  SEARCH /W

* iNDX WILL NOW LOOK FOR CUSPS IN FOLLOWING ORDER :

  1) SEARCH PATH
  2) ROOT DIRECTORY
  3) LOGICAL PATHNAME OF ' ' (NULL)

  LOOKS FOR .86 FILE IN ALL THREE THEN IF NOT FOUND SEARCHES FOR NO EXTENSION IN ALL THREE.

TIP :

IF USERS HAVE LOCAL WINCHESTER STORAGE, SEARCHING THE LOCAL WINI FOR SYSTEM FILES WILL INCREASE NETWORK PERFORMANCE.
ENSURE THAT ALL SYSTEM CUSPS AND COMMONLY USED UTILITIES ARE ON THE LOCAL WINI. PUT THE COMMAND : SEARCH /W IN YOUR INIT.CSD FILE AND EVERYTHING WILL GO FASTER FROM THERE.

SUPERUSER MAY CLEANUP THE ROOT DIRECTORY BY PUTTING ALL THE SYSTEM FILES (.86) IN A DIFFERENT DIRECTORY AND ALL USERS SETTING THEIR SEARCH PATH TO THE SYSTEM DIRECTORY IN THEIR INIT.CSD FILE.

RELEASE 2.5

NEW CUSPS

NETSTAT : iNDX CUSP THAT OPERATES ON NRM

* SHOWS LOG STATE OF EACH WORKSTATION

```
NETSTAT
iNDX-NII (V2.6) NETSTAT VERSION 2.6
     NETWORK STATUS : LOGON ENABLED
     WORKSTATION STATUS :
NODE NO.   LOCATION     STA.TYPE    LOG STATE     DJC STATUS
   1       BRIAN'S OFF  SIV/4         HANK          PRIVATE
   2       CHRIS'S OFF  SIV/4       INACTIVE
   3       CLUSTER      SIII          FAE          PRIVATE
```

RELEASE 2.5

NEW CUSPS

* NETDOWN - DISABEL LOGON TO NDS-II, OPERATES ON NRM

  - PRINTS MESSAGE SAYING LOGON DISABLED.

    SERIES-IV :

        NETDOWN - LOGON DISABLED

    SERIES-II OR CLUSTER :

        NO MESSAGE.

  - ALL LOGONS DISABLED, IF USERS ARE LOGGED ON THEN THEY AREN'T EFFECTED UNTIL THEY LOGOFF AND THEN TRY TO LOGON.

  - MESSAGE PRINTED IF USER TRIES TO LOGON IF LOGON DISABLED :

    SERIES-IV :

        LOGON DISABLED

    SERIES-II :

        NETWORK FILE DETACHED, DEVICE DISMOUNTED

    CLUSTER :

        ERROR 80

NETDOWN CAN BE USED TO DISABLE LOGONS DURING AN VERIFY, ETC. IT GUARANTEES THAT ONCE NETDOWN IS EXECUTED THEN NO USER WILL ACCESS TO NETWORK UNTIL NETUP IS ACTIVATED.

RELEASE 2.5

NEW CUSPS

* NETUP - ENABLE LOGON TO NDS-II, OPERATES ON NRM

- PRINTS MESSAGE SAYING LOGON ENABLED.

SERIES-IV :

NETUP - LOGON ENABLED

SERIES-II OR CLUSTER :

NO MESSAGE

NETUP IS USED TO REENABLE LOGONS AFTER A SYSTEM CUSP(ARCHIVE) HAVE EXECUTED A NETDOWN DUE TO CERTAIN ERROR CONDITIONS. PREVIOUSLY TO REENABLE LOGONS THE NRM WOULD HAVE TO BE RESET. NETUP IS ALSO USED TO REENABLE LOGONS AFTER THE SUPERUSER HAS EXECUTED A NETDOWN.

NEW CUSPS

* NETMSG - SEND MESSAGE TO ALL SERIES-IV USERS, OPERATES ON NRM

  WILL SEND MESSAGE IMMEDIATELY, EVEN IF USERS ARE USING AEDIT.
  THIS WILL NOT AFFECT FILE, BUT SCREEN REFRESH IS REQUIRED.

  EXAMPLE :

```
NETMSG
LOGOFF PLEASE, NETWORK COMING DOWN
1 MINUTE TIL RESET
```

  THE . STARTING IN COLUMN 0 IS THE END OF MESSAGE MARKER.

RELEASE 2.5

NEW CUSPS

* NETMAP - ISIS-III(N) CUSP.

ALLOWS UP TO 9 REMOTE DEVICE ASSIGNMENTS, REGARDLESS OF THE PHYSICAL CONFIGURATION OF LOCAL STORAGE.
-NETMAP 8 ;ALLOWS 8 REMOTE DEVICE ASSIGNMENTS

NETMAP SOLVES THE PROBLEM OF ONLY BEING ABLE TO ASSIGN 5 REMOTE DEVICES ON A SERIES-II WITH 720 DRIVES.

RELEASE 2.5

OTHER NEWS

* :WORK: AUTOMATICALLY ASSIGNED AT LOGON TO HOME DIRECTORY.

* LNAME LIMIT INCREASED TO 10

* HOME DIRECTORY NOW ALLOWED FOR SUPERUSER

  - SUPERUSER MAY NO LOGON TO A CLUSTER

* ACCESS NOW ACCEPTS SAME ENTRIES FOR DATA/DIR FILES

* COMPLETE SERIES-IV LIKE MENU LINE NOW ON NRM CONSOLE

* COPY/DELETE NOW ACCEPTS CONCATENATED FILES

* FILL NOW CAN BE USED IN A SUBMIT FILE

* SYSGEN CHANGES :

  - DEFAULT NETWORK ACCESS RIGHTS
  - OVERLAY CACHE : SET AMOUNT OF SYSTEM OVERLAYS IN MEMORY.

    OPTIMAL NUMBERS ARE :

      - 5 FOR SERIES IV/3, 7-10 FOR SERIES IV/4
      - SHOULD BE 0 FOR NRM

  - SPU/CPIO TRACE : DIAGNOSTIC TRACES MAY BE SET FOR MEMORY DUMPS.
    SHOULD BE SET TO 0.

* WILDCARD HANDLING

- *.* WILL PARSE TO ALL FILES (ISIS COMPATIBLE)
- * WILL PARSE TO ONLY FILES WITHOUT EXTENSION (ISIS COMPATIBLE)
- A* WILL PARSE TO ALL FILES THAT START WITH A
- A*.* WILL PARSE TO ALL FILES THAT START WITH A AND HAVE A '.' IN THE FILE NAME

* TAPE BACKUP AVAILABLE

RELEASE 2.8

WHAT TO EXPECT

* TWO USER SERIES-IV
  - FOREGROUND/BACKGROUND
  - SINGLE USER WITH 2 PARTITIONS (LOGGED ON TWICE)
  - TWO USERS WITH EQUAL OR UNEQUAL PRIORITY
* SERIES-IV CLUSTER
* SYSGEN CHANGES
  - DEFAULT SERIES-IV CONFIGURATION
  - DEFAULT NRM CONSOLE CONFIGURATION
* BUG FIXES
* EXPANDED SYNTAX GUIDE

RELEASE 2.5

VERSION NUMBERS

iNDX.GII	V2.6
iNDX.BOOT	V2.5
NRM.CUSPS	V2.6
DIAGNOSTICS	V2.5
iNDX.W31	V2.6
iNDX.W41	V2.6
iNDX.S31	V2.6
iNDX.S41	V2.6
ISIS-III(N)/(C)	V2.5

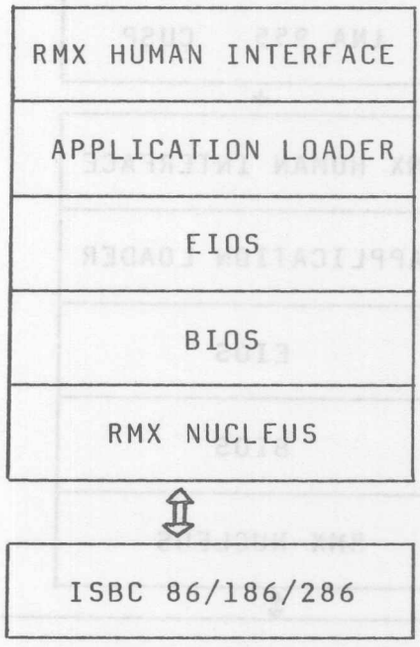
RMX/NDS-II LINK

WHY ?

* YOU ASKED FOR IT
* IMPROVES SUPPORT FOR RMX DEVELOPMENT
* RMX LINK AT ETHERNET SPEEDS
* VALUE ADDED TO NDS-II
* SUPPORTED ON :
  - 310, 330 AND 380 SYSTEMS
  - RMX 86 AND RMX 286R

RMX/NDS-II LINK

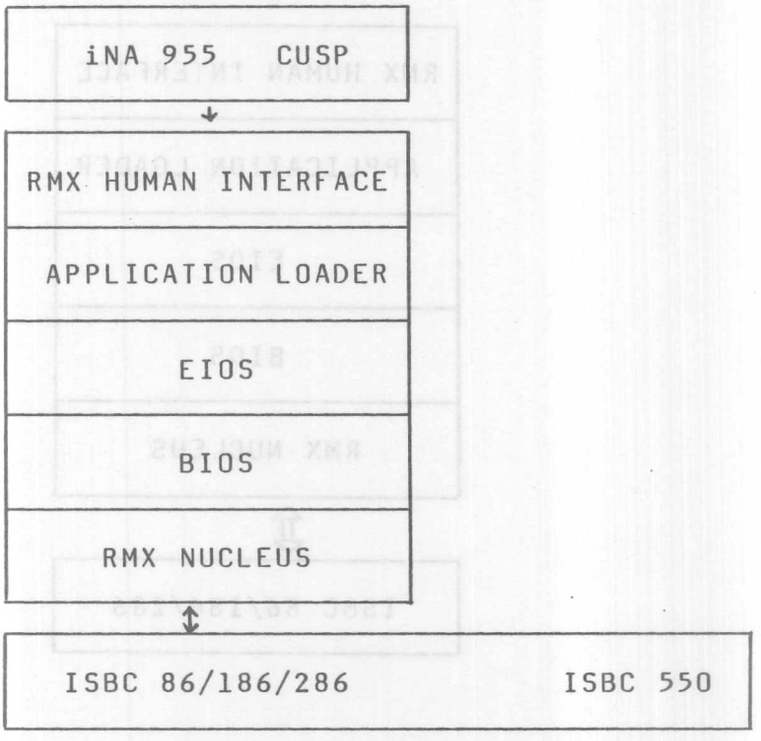
RMX OPERATING SYSTEM



INA 955 FUNCTIONAL DIAGRAM

RMX/NDS-II LINK

RMX + INA 955

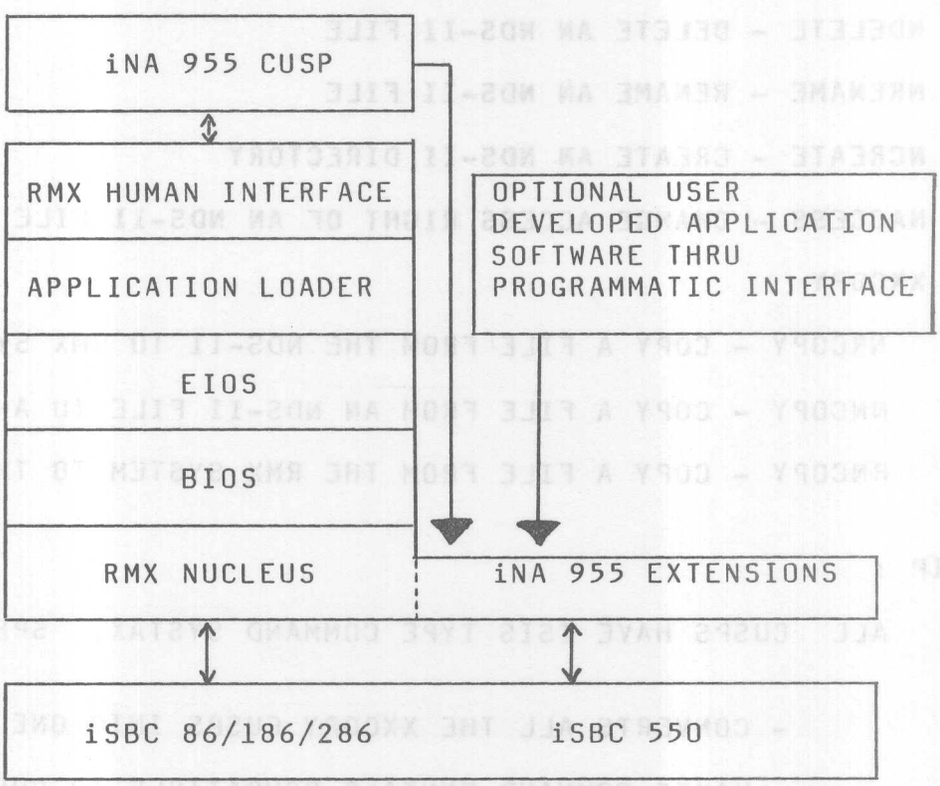


iNA 955 FUNCTIONAL DIAGRAM

HOW IT COULD HAVE BEEN DONE

RMX/NDS-II LINK

RMX + iNA 955



iNA 955 FUNCTIONAL DIAGRAM

HOW IT WAS DONE USING OS EXTENSIONS

RMX/NDS-II LINK
CUSPS

* NLOGON - LOGON USER TO NDS-II
* NLOGOFF - LOGOFF USER FROM NDS-II
* NDIR - PRINT A DIRECTORY LISTING OF AN NDS-II DIRECTORY
* NDELETE - DELETE AN NDS-II FILE
* NRENAME - RENAME AN NDS-II FILE
* NCREATE - CREATE AN NDS-II DIRECTORY
* NACCESS - CHANGE ACCESS RIGHT OF AN NDS-II FILE
* XXCOPY

  NRCOPY - COPY A FILE FROM THE NDS-II TO RMX SYSTEM

  NNCOPY - COPY A FILE FROM AN NDS-II FILE TO AN NDS-II FILE

  RNCOPY - COPY A FILE FROM THE RMX SYSTEM TO THE NDS-II

TIP :

ALL CUSPS HAVE ISIS TYPE COMMAND SYSTAX. SPECIAL THANKS TO

- CONVERTS ALL THE XXCOPY CUSPS INTO ONE CALL
- MAKES COMMAND SYNTAXS COMPATIBLE TO CURRENT RMX CUSPS

RMX/NDS-II LINK

iNA 955

* 10 NEW CUSPS PROVIDED

* SYSTEM LIBRARY PROVIDED WITH 13 NEW OS CALLS (OS EXTENSIONS)

* LARGE, MEDIUM AND COMPACT LIBRARIES PROVIDED

* CONFIG FILE SUPPLIED TO CONFIGURE iNA 955 INTO RMX SYSTEM

* CSD FILE SUPPLIED TO LINK AND LOCATE iNA 955

* SUPPLIED ON RMX FORMAT DISK 8 INCH SSSD AND 5-1/4 INCH

RMX/NDS-II LINK

POINTS TO REMEMBER

* ISIS-DFS LOOKALIKE

* ALL FILENAMES MUST BE ISIS FILENAMES

* CANNOT HAVE MORE THAN ONE RMX USER PER RMX SYSTEM LOGGED ON THE NDS-II AT ONE TIME

* ONLY SIX OPEN FILES ALLOWED

* MUST LOGON BEFORE ANY FILE TRANSFER CAN BE DONE

* PATHNAME LIMIT OF 128 CHARACTERS

* USER MUST MAINTAIN FILE POINTER ON ALL FILE I/O OS CALLS

* FILE COPY ONLY

* NO PMTs OR ELECTRONIC MAIL

EXEMPLE :

```
NLOGON
     USERNAME - FRED
     PASSWORD -
RNCOPY FILE1.LST TO FILE2.LST
NLOGOFF
```

RMX/NDS-II LINK

OS CALLS

FILE I/O CALLS :

NQ$OPEN

NQ$CLOSE

NQ$READ

NQ$WRITE

NOTE :

THE USER MUST KEEP TRACK OF THE FILE POINTER

SYSTEM CALLS :

NQ$CHANGE$ACCESS

NQ$CREATE$DIR

NQ$DELETE

NQ$FILE$INFO

NQGETVIRTUAL$ROOT

NQ$LOGOFF

NQ$LOGON

NQ$READ$DIR$EXPANDED

NQ$RENAME

NOTE :

ALL CALLS ARE OS EXTENSIONS AT THE NUCLEUS LEVEL

ELECTRONIC MAIL

* MAIL IS AN IMPORTANT ADDITION TO THE NDS-II NETWORK

* REQUIRES iNDX V2.5 OR GREATER

* FREE UPDATE

* USERS CAN SEND MESSAGES TO :
  - OTHER USERS
  - GROUPS
  - BULLETIN BOARDS

* MESSAGES MAY BE :
  - INTERACTIVE TEXT
  - A TEXT FILE
  - AN OBJECT FILE

* USERS CAN READ MESSAGES FROM :
  - THEIR OWN MAILBOX
  - GROUP MAILBOXES IF THEY ARE A MEMBER OF THE GROUP
  - BULLETIN BOARDS

* COMMAND SET IDENTICAL BETWEEN 8 AND 16 BIT VERSIONS

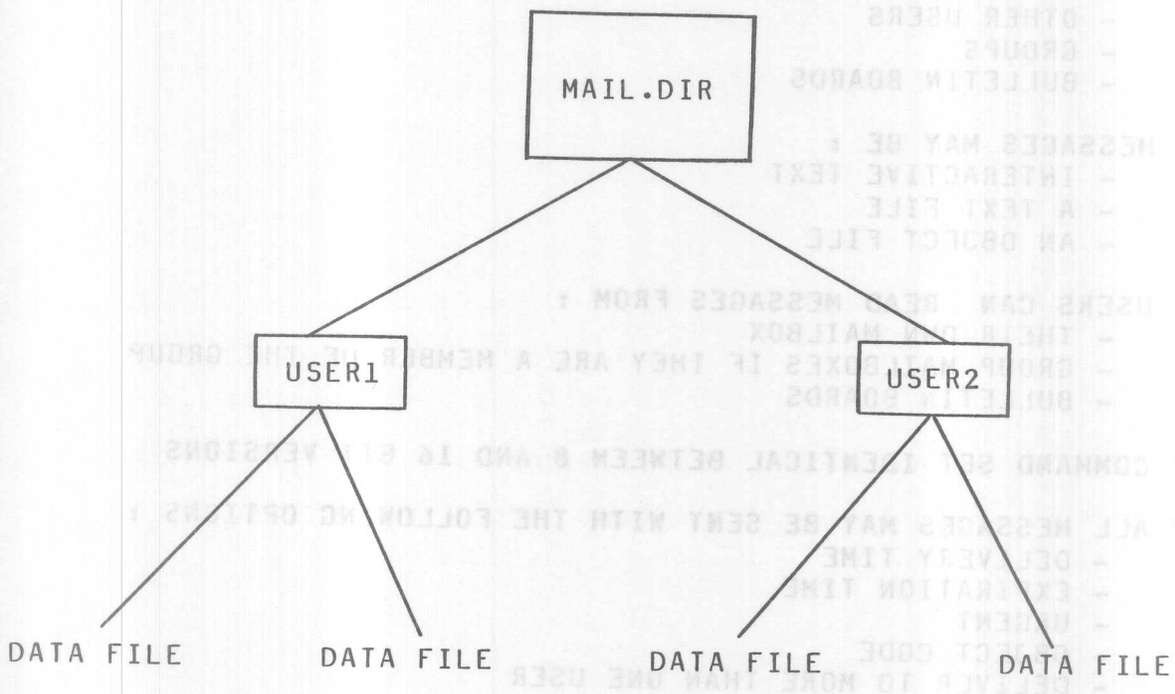
* ALL MESSAGES MAY BE SENT WITH THE FOLLOWING OPTIONS :
  - DELIVERY TIME
  - EXPIRATION TIME
  - URGENT
  - OBJECT CODE
  - DELIVER TO MORE THAN ONE USER

* RECEIVED MESSAGES MAY BE :
  - DELETED AFTER THEY ARE READ
  - SAVED TO A FILE
  - SAVED IN AN ALTERNATE MAILBOX
  - SAVED TO :SP:
  - FORWARDED TO OTHER USERS
  - REPLIED TO DIRECTLY

* LIMITATIONS :

  - NO NRM TO NRM MAIL
  - NO NOTIFICATION OF RECEIVED MESSAGE, MUST CHECK MESSAGES

ELECTRONIC MAIL



ELECTRONIC MAIL

* ABILITY TO MAIL MESSAGES, SOURCE FILES OR OBJECT FILES EITHER INTERACTIVELY OR NONINTERACTIVELY

  - INTERACTIVE MODE

```
MAIL
*MAIL TO BILL, GROUP_240, MARY
=HI,
=HOW ARE YOU TODAY
=E                          ;NOTE HOW YOU EXIT WITH E
*EXIT
```

  - NONINTERACTIVE MODE

```
MAIL FILE.OBJ TO FRED,MARY  OBJECT
```

THE '*' IS THE INTERACTIVE MODE PROMPT AND THE '=' MESSAGE INPUT PROMPT.

MAIL.TMP FILE CREATED FOR EACH MAIL SESSION.

THE URGENT OPTION WILL FLAG THE MESSAGE AS URGENT.

TIP :

```
MAIL
*MAIL TO BRAIN
=THIS IS A MAIL SESSION
=E
BRAIN : BOX DOESN'T EXIST
*MAIL :F9:MSG.TMP TO BRIAN
```

IF ERROR WHEN MAILING MESSAGE, YOU CAN TRY AGAIN BY USING THE MAIL.TMP FILE. :F9: REQUIRED ONLY IN ISIS.

* ABILITY TO CHECK YOUR MAIL BOX, GROUP BOX OR BULLETIN BOARDS AT ANYTIME

MAIL USES A CURRENT MAILBOX FOR ALL COMMANDS. THIS BOX IS YOUR PERSONAL BOX (IE. YOUR LOGON USERNAME) UNLESS OTHERWISE SPECIFIED WITH THE BOX OPTION.

- INTERACTIVE

```
MAIL
NO MESSAGES
*EXIT
```

- NONINTERACTIVE

```
MAIL CHECK BOX SIV_NEWS
          1   JOHN        1/12/84     PPC MEETING UPDATE
    *     2   GERI        1/23/84
    +     3   ARNIE       2/2/84      TREE LINK FILE

 * URGENT
 + OBJECT
```

THE BOX OPTION ALLOWS A USER TO CHECK GROUP MAILBOXES OR BULLETIN BOARDS.

EXEMPLE :

```
MAIL            ; A CHECK IS AUTOMATICALLY DONE
*BOX APS        ; THE CURRENT MAILBOX IS SWITCHED TO APS AND A
                ; CHECK IS AUTOMATICALLY DONE
```

ELECTRONIC MAIL

* EACH USERNAME HAS THEIR OWN MAIL BOX

EITHER THE SUPERUSER CAN CREATE ALL THE MAILBOXES WITH THE USER COMMAND OR IF A USER ACCESSES MAIL FOR THE FIRST TIME A MAILBOX WILL BE CREATED

YOU CAN'T SEND MAIL TO A USER UNTIL THEY HAVE A MAILBOX.

```
MAIL
*LIST
    BILL
    FRED
    SIV_NEWS
    NRM_NEWS
*LIST BOARD
    SIV_NEWS
    NRM_NEWS
*E
```

THE LIST COMMAND WILL LIST ALL USER MAILBOXES, GROUP BOXES AND BULLETIN BOARDS :

```
    LIST USER        ; LISTS ONLY THE USER MAILBOXES.
    LIST GROUP       ; LISTS ONLY THE GROUP BOXES.
    LIST GROUP NAME  ; LISTS USERS IN GROUP NAME.
    LIST BOARD       ; LISTS ONLY THE BULLETIN BOARDS
    LIST             ; LIST ALL BOXES
```

ANY USER CAN SEND MAIL TO ANY MAILBOX BUT THE READING OF THE MAILBOX IS DEPENDENT ON :

    USER MAILBOX - ONLY THE OWNER OF THE BOX CAN READ IT
    GROUP MAILBOX - ONLY THE USERS OF THE GROUP CAN READ IT
    BULLETIN BOARD - ANYONE CAN READ IT

USERS MAILBOXES CAN ONLY BE DELETED BY THE SUPERUSER USING THE REMOVE COMMAND

```
* USERS MAY ALSO BELONG TO GROUP MAILBOXES
  LOGON BRIAN
  MAIL
  *GROUP ENGIN (JOHN,CHRIS,MARY)
  *EXIT

  CREATES A GROUP CALLED ENGIN AND TELLS MAIL THAT BRIAN, JOHN,
  CHRIS AND MARY BELONG TO THE GROUP.
  ADDING USERS TO EXISTING GROUPS :

  - INTERACTIVE
    LOGON BRIAN
    MAIL
    *BOX ENGIN ;YOUR CURRENT BOX MUST BE SET TO THE GROUP BOX
    *ADD ANN

  - NONINTERACTIVE
    LOGON BRIAN
    MAIL ADD ENGIN (ANN)

  TIP :
    HOW TO HAVE MULTIPLE MAILBOXES OF YOUR OWN :
          LOGON BRIAN
          MAIL
          *GROUP BRIANS_BOX
               ; A GROUP IS CREATED WITH THE NAME OF "BRIANS_BOX"
               ; AND BRIAN IS THE ONLY MEMBER
    TO CHECK MORE THAN ONE BOX :
          MAIL
               ; MESSAGE SUMMARY PRINTED
          *BOX BRIANS_BOX
               ; MESSAGE SUMMARY PRINTED
          *EXIT
               OR  (IN YOUR INIT.CSD FILE)
          MAIL
          BOX BRIANS_BOX
          EXIT
```

* THE CREATOR OF THE GROUP BOX IS THE GROUP BOX OWNER.
  COMMANDS AVAILABLE ARE :

WHICH - SHOWS WHICH USER HAS DELETED WHICH MESSAGES IN GROUP BOX

```
LOGON BRIAN
MAIL
*BOX ENGIN
*WHICH (+) JOHN
     4 SUSIE      2/2/84
     6 WINTHROP   2/22/84
```

JOHN "HASN'T" DELETED MESSAGE 4 OR 6. IF THE '+' OPTION IS USED TO PRINT ONLY THE MESSAGES JOHN "HAS" DELETED WILL BE LISTED.

WHO - SHOWS USERS THAT HAVE DELETED A PARTICULAR MESSAGE

```
LOGON BRIAN
MAIL
*BOX ENGIN
     ; MESSAGE SUMMARY PRINTED
*CR
     ; FIRST MESSAGE IN LIST WILL BE PRINTED
*WHO (+) MESSAGE $
     JOHN
     MARY
```

JOHN AND MARY "HAVE" DELETED MESSAGE 1. THE '+' OPTION IS USED TO PRINT USERS WHO "HAVEN'T" DELETED THE MESSAGE. A MESSAGE NUMBER MAY BE SPECIFIED AND THE DATA FOR THE MESSAGE SPECIFIED WILL BE PRINTED INSTEAD OF THE CURRENT MESSAGE.

DROP JOHN,MARY - REMOVE JOHN AND MARY FROM GROUP

```
LOGON BRIAN
MAIL
*BOX ENGIN
*DROP JOHN,MARY
```

REMOVE GROUP_NAME - REMOVES THE GROUP BOX FROM THE SYSTEM

```
LOGON BRIAN
MAIL
*REMOVE ENGIN
```

ONLY THE GROUP OWNER CAN REMOVE MESSAGES FROM THE GROUP BOX.

* ABILITY TO SET TIME AND DATE FOR WHEN A MESSAGE IS TO BE DELIVERED OR HOW LONG A MESSAGE MAY STAY IN A MAILBOX BEFORE MAIL AUTOMATICALLY DELETES IT.

MAIL TO CHARLIE DELIVER (11/28) EXPIRE (1/2/85, 12:00:30)
= THOS IS A DELAYED MESSAGE AND WILL EXPIRE
= E

THESE OPTIONS ALLOW A USER TO DELAY THE DELIVERY OF A MESSAGE TO A BULLETIN BOARD, GROUP OR USER OR HAVE A MESSAGE REMOVED AUTOMATICALLY AT A CERTAIN DATE IF THE MESSAGE IS TIME CRITICAL.

ELECTRONIC MAIL

* BULLETIN BOARDS MAY BE SET ON YOUR SYSTEM.

ANY USER CAN READ FROM AND WRITE TO A BOARD BUT ONLY THE OWNER OF THE BOARD CAN DELETE MESSAGES.

- CREATING A BOARD

```
MAIL
*BOARD WIZZO.PROGRAMS
*EXIT
```

- READING A BOARD

```
MAIL
*BOX WIZZO.PROGRAMS
     ;DISPLAYS MESSAGES AND SETS DEFAULT BOX TO
     ;WIZZO.PROGRAMS
```

ONLY THE BOARD OWNER MAY REMOVE THE BOARD IT IS A GOOD IDEA TO USE THE EXPIRE OPTION. (OR PURCHASE ANOTHER 84MB DRIVE !).

ALWAYS USE THE SUBJECT OPTION WHEN SENDING A MESSAGE

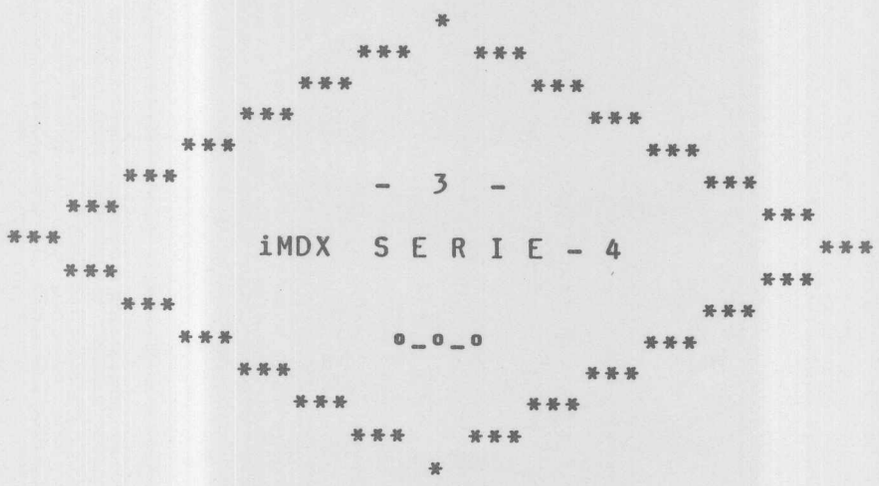
ELECTRONIC MAIL

```
*ANY MESSAGE RECEIVED MAY BE SAVED TO A FILE OR DELETED FROM YOUR BOX
  MAIL
   1      JOHN      2/2/84
   2      BRIAN     2/10/84
 *CR                ; WILL DISPLAY MESSAGE 1
 *PRINT 2           ; WILL DISPLAY MESSAGE 2
 TO DELETE A MESSAGE ENTER :
           *DELETE    ; WILL DELETE THE CURRENT MESSAGE
           *DELETE 2 ; WILL DELETE MESSAGE 2
 TO SAVE A MESSAGE ENTER :
           *SAVE MAIL.SAV ; WILL SAVE THE CURRENT MESSAGE AS MAIL.SAV
           *SAVE 2 MAIL.SAV ; WILL SAVE MESSAGE 3 AS FILE MAIL.SAV
 TO REPLY TO THE SENDER OF THE CURRENT MESSAGE :
           *REPLY
           =THIS IS A REPLY
           =E
                 OR
           *REPLY 2   ; WILL REPLY TO MESSAGE 3
 TO FORWARD A MESSAGE ENTER :
           *FORWARD TO JOHN,WAYNE SUBJECT("PUT SUBJECT HERE")
                ; WILL FORWARD THE CURRENT MESSAGE
           *FORWARD 2 TO JOHN,WAYNE SUBJECT("PUT SUBJECT HERE")

 THE  CURRENT  MESSAGE IS THE MESSAGE LAST READ,  TO  CHANGE  THE
 CURRENT  MESSAGE USE THE PRINT COMMAND OR KEEP READING  MESSAGES
 IN ORDER.

 THE SUBJECT STRING CAN BE 48 CHARACTERS OR LESS.

 MAIL HAS TWO MODES OF EXIT.
     - EXIT - EXIT AND UPDATE MAILBOX
     - QUIT - EXIT WITH NO UPDATE
```


- 3 -
iMDX S E R I E - 4

L'outil de développement iMDX SERIE - 4 est organisé autour de l'Opérating System iNDX. Il suit donc la même évolution que le NRM de réseau NDS - 2.

De la même façon la nouvelle version 2.5 pour le SERIE - 4 est disponible et en Août la version 2.8 sera proposée.

Voici les caractéristiques importantes des nouvelles fonctionnalités.

SERIES-IV NEW CUSPS

(Version 2.5)

SEARCH :

. THE SEARCH COMMAND ALLOWS YOU TO SPECIFY AN OPTIONAL SEARCH PATH FOR CUSPS. RULES ARE :

- OPTIONAL SEARCH PATH FOR FILE.86
- MAIN VOLUME FOR FILE.86
- NULL LOGICAL PATHNAME FOR FILE.86
- OPTIONAL SEARCH PATH FOR FILE
- MAIN VOLUME FOR FILE
- NULL LOGICAL PATHNAME FOR FILE
- FILE NOT FOUND

. SEARCH (PATHNAME) IS HANDY FOR TUCKING COMPILERS, ETC... AWAY IN SUB-DIRECTORIES.

. SEARCH (LOCAL WINI) IS BEST FOR IMPROVING PERFORMANCE WHEN NETWORK IS LOADED DOWN ! NO CONTENTION/ETHERNET LOAD TIME WHEN LOADING AEDIT, COMPILERS, ETC...!

SITTY :

. THE SITTY COMMAND IS USED FOR THE FOLLOWING FONCTIONS :

- SET BAUDRATE OF SERIAL CHANNEL 1 ON SERIES-IV,
  BAUDRATE OF MAIN CONSOLE AT NRM.

- ENABLE/DISABLE DISPLAY OF TYPE-AHEAD BUFFER

- UPDATE TERMINAL CONFIGURATION - RE-MAP KEYS, ETC.
  (WE FIND THIS USEFUL FOR SETTING SERIES-IV
  UNUSED KEYS TO USEFUL FUNCTIONS)

- REMOTE/LOCAL : SET CONSOLE TO SERIAL CHANNEL 1
  AND BACK - THIS IS A VERY NICE FEATURE !
  CONNECTED TO A MODEM, ONE CAN WORK REMOTELY WITH
  COMPLETE CONTROL OF THE SERIES-IV - FROM HOME,
  OR WITH SERIES-IV IN NOISY LAB, QUIET TERMINAL
  IN OFFICE !

NETSTAT :

THE NETSTAT SHOWS STATUS OF STATIONS ON NETWORK

NETMSG :

NETMSG CAN SEND MESSAGES FROM NRM TO ALL SERIES-IVS AND FROM SERIES-IV TO NRM AND ALL OTHER SERIES-IVS ON NET. MESSAGE BROADCASTS ON USERS SCREEN, REGARDLESS OF CURRENT OPERATION.

NETUP/NETDOWN :

TAKE THE NETWORK AND UP, NETDOWN GRACEFULLY SHUTS DOWN NETWORK BY PREVENTING FURTHER LOGONS, ALLOWS CURRENT USERS TO FINISH WORK AND LOGOFF.

SERIES-IV FUTURE CAPABILITES

(VERSION 2.8)

TWO USER :

- SECOND USER ATTACHED VIA SERIAL CHANNEL 1
- CAPABILITY AVAILABLE ON STANDALONE OR NETWORK SYSTEMS
- SECOND USER USES THE BACKGROUND PARTITION
- ONLY ONE USER CAN RUN ISIS-IV, BUT...
- BOTH USERS ARE iNDX !

REGION :

- SETS UP OPERATING MODE, PARTITIONS, TWO USERS

- NEW "TOGGLE" OPTION GIVES SINGLE USER INTERACTIVE ACCESS TO BOTH FOREGROUND AND BACKGROUND PARTITIONS ! DO TWO INTERACTIVE JOBS AT ONCE - DISPLAY A DIRECTORY WHILE IN THE MIDDLE OF AN EDIT SESSION, AND MANY MORE USES !

- NOW SETS PARTITIONS IN 1K "PAGES", NOT PARAGRAPHS

- SETS ISIS-IV PARTITION TO FOREGROUND OR BACKGROUND USER

- SETS SYSTEM PRIORITY BETWEEN USERS - HIGHER, LOWER, EQUAL

- DISPLAYS CONSOLE STATUS - WHO HAS IT, OR IS SECOND ONE THERE.

SYSGEN :

. SYSGEN WILL BE EXPANDED IN RELEASE 2.8 TO INCLUDE :

- TERMINAL BAUDRATE FOR SECOND TERMINAL
- SEARCH PATH FOR TERMINAL CONFIG.FILE
- NEW REGION CUSP OPTIONS INCLUDING TWO USER OR TOGGLE, PRIORITY, ETC.

. WHEN THESE OPTIONS ARE SET VIA SYSGEN, THEY NO LONGER NEED BE SPECIFIED IN THE INIT.CSD. BOOT-UP WILL BE MUCH QUICKER - RUNNING REGION, STTY, ETC CAN STILL BE DONE, SYSGEN MERELY PROVIDES A MUCH MORE FLEXIBLE DEFAULT.

. OVERLAY CACHE (2K PER) - ENHANCES CUSPS WITH OVERLAYS

. DIAGNOSTIC TRACE MODE (FOR PROBLEM SYSTEMS ONLY, DEGRADES PERF.)

. FUTURE SYSGEN OPTIONS (REL.2.8.) INCLUDE :

- TERMINAL BAUDRATE, SEARCH PATH FOR TERMINAL CONFIG.FILE
- NEW REGION CUSP OPTIONS - NO LONGER NEEDED IN INIT.CSD !
- 2-USER OR TOGGLE MODE, ETC.

SERIES-IV FUTURE CAPABILITIES (CONT.)

ISIS CLUSTER :

. INDX RELEASE 2.8 WILL PROVIDE ISIS CLUSTER SUPPORT FOR SERIES-IV.

- UP TO FOUR CLUSTERS CAN BE ADDED TO THE SERIES-IV.
- WITH TWO-USER CAPABILITY, MAXIMUM USERS PER SERIES-IV IS SIX.
- ISIS CLUSTER REQUIRES NDS-II, NO STANDALONE SERIES-IV.

. CLUSTER "GOTCHAS" INCLUDE :

- EACH CLUSTER TAKES 64K PAGE OF SYSTEM MEMORY, STARTING FROM (TOPOFMEM-MONITOR) AND GROWING DOWNWARD
- 012B HALF-MEG MEMORY BOARD WILL BE STRAPPED SHORT IN SOME CASES TO ACCOMODATE CLUSTERS.

MENU COMPILER :

. THE MENU COMPILER IS HOW THE MENU LINE, SYNTAX GUIDE, AND ON-LINE HELP COMMANDS ARE CREATED FOR SERIES-IV/NRM.

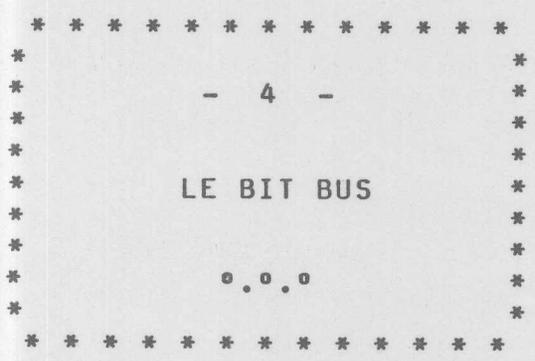
. THIS COMPILER IS EVENTUALLY TO BE EXTERNALIZED TO THE FIELD.

. CUSTOMERS WILL BE ABLE TO CUSTOMIZE THE MENU LINE AND HELP FILES TO THEIR LIKING :

- REMOVE UNWANTED COMMANDS
- ADD CUSTOMIZED CUSPS
- PROVIDE DESCRIPTIVE HELP TEXT, OR REMOVE IT.

. COMPILER WILL NOT BE TIED TO ANY FORMAL RELEASE OF SOFTWARE

. FIRST RELEASE WILL BE CONDUCTED THROUGH APPS. FOR CONTROL.


- 4 -
LE BIT BUS

LE BIT BUS EST UN BUS SERIE DEFINI PAR iNTEL DONT LE BUT EST DE PERMETTRE L'INTERCONNEXION D'UNE FAMILLE DE PRODUITS ORIENTES VERS LES ACQUISITIONS D'ENTREE/SORTIE INDUSTRIELLES. LE PROTOCOLE DE COMMUNICATION SUPPORTE PAR LE BIT BUS PERMET D'ASSURER DES ECHANGES DE RESSOURCES DANS UNE ARCHITECTURE DECENTRALISEE.

LE BIT BUS ETANT UNE LIAISON SERIE, LE COUT DE LA CONNEXION EST TRES FAIBLE TANDIS QUE LE PROTOCOLE DU TYPE SDLC OPTIMISE LE CRITERE VITESSE D'ECHANGE.

AFIN DE FACILITER LA MISE EN OEUVRE D'UNE APPLICATION DES OUTILS MATERIELS DE BASE SONT PROPOSES (RCB 44/10 - SBX 344) AINSI QU'UN MONITEUR TEMPS REEL (RMX 51) QUI S'INTERFACE DIRECTEMENT A RMX 86.

POUR LE MILIEU INDUSTRIEL L'EXISTENCE DE CETTE NOUVELLE GENERATION DE PRODUITS DOIT ETRE TRES IMPORTANTE ET C'EST POURQUOI DES SPECIFICATIONS RELATIVEMENT COMPLETES SONT A LIRE AVEC ATTENTION.

Table of Contents

1 Introduction
Page iii

2 iSBX™ 344 BITBUS™ Controller MULTIMODULE™ Board
Page 1

3 iRCB 44/10 BITBUS™ Remote Controller Board
Page 11

4 iRMX™ 51 Real-Time Multitasking Executive
Page 21

5 iRMX™ 510 iDCM Support Package
Page 27

6 8084AH/8344AH High Performance Microcontroller with On-Chip Communication Center
Page 31

7 BITBUS™ Interconnect Serial Control Bus Specification
Page 51

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Intel Corporation.

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BITBUS	iLBX	iPDS	Plug-A-Bubble
COMMputer	i_m	iPSB	Prompt
CREDIT	iMMX	iRMX	Promware
Data Pipeline	Insite	iSBC	QUEX
GENIUS	Int_el	iSBX	QUEST
$\overset{\Delta}{i}$	int_el	iSDM	Ripplemode
i	int_elBOS	iSSB	RMX/80
I^2ICE	Intelevision	iSXM	RUPI
ICE	$Int_eligent$ Identifier	Library Manager	Seamless
iCS	$Int_eligent$ Programming	MCS	SOLO
iDBP	Intellec	Megachassis	SYSTEM 2000
iDIS	Intelink	MICROMAINFRAME	UPI
	iOSP	MULTIBUS	

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

Copyright© 1984. Intel Corporation

Introduction to the Distributed Control Modules

Overview

Intel's Distributed Control Module (iDCM) products provide building blocks for construction of real-time distributed control systems based on the BITBUS™ interconnect. This new serial bus architecture addresses many of the limitations inherent in traditional connection methods. For instance, future system cost reductions are limited in systems based on parallel bus structures due to their electrical and mechanical characteristics. Other traditional connection methods such as current loops and RS 232 C do not provide sufficient performance or flexibility for complex industrial control applications. In addition, although there are numerous industry standards for connecting microprocessors, the MULTIBUS® and the STD-bus for example, there is no standard connection for microcontrollers. The BITBUS interconnect (Table 1) combines existing standards with new standard interfaces to provide the optimal solution for difficult distributed control problems.

The iDCM products combine hardware and software for use in applications that would benefit most from employing distributed architectures. Applications such as robotics, process control, data acquisition and control, and environmental control are a few examples.

Table 1. STANDARD BITBUS™ INTERFACES

Interface	Specification
Electrical	RS485
Cable	10-conductor flat ribbon or 1 to 2 wire twisted pair
Back-plane connector	64-pin Standard DIN
End-cable connector	3M #3446-1302 female
Control-board form-factor	Single-height, Double-depth Eurocard
Data Link control	Synchronous Data-link Control
Data transfer rate	62.5K baud, 375K baud and 2.4M baud
Message formats	Compatible with iRMX 51 format command/response/status
Common command sequences	Integral Remote Access and Control (RAC) function
Operating systems	S/W drivers for iRMX 86, 88, 286R and ISIS (for iPDS only)

Benefits of Distributed Architectures

Distributed architectures are intrinsically more reliable than centralized architectures. In a centralized control system a central controller failure results in a system-wide failure. Distributed systems can be configured to prevent this. Also, distributed systems are more cost effective and more easily modified. For instance, performance improvements in centralized systems are expensive and do not concentrate improvements in the areas where they are needed most. In distributed systems, only the specific parts of the system that require enhancement need be modified. Most importantly, control systems based on distributed architectures have less difficulty responding to the external environment because they have less to manage.

The BITBUS™ Interconnect

The BITBUS interconnect is a serial bus optimized for high speed transfer of short control messages in a hierarchical system. In order to provide an easy to use high performance serial interconnect, transparent to the applications programmer, high-level interfaces are specified. These interfaces include: the message structure and protocol for a multitasking environment, and a set of high-level commands for remote I/O access and application task control. As with traditional bus specifications, the electrical and data protocol levels have been defined.

The BITBUS interconnect supports up to 250 nodes and three bit rates dependent on application performance requirements. Different BITBUS segments may support different bit rates.

A Simple and Reliable Solution

The BITBUS architecture supplies the system designer with a simple and reliable foundation. Some key features of this architecture are: defined high-level interfaces that provide all communication and user program management, the reliable SDLC protocol, power-up diagnostics, standard industrial packaging, compact software and hardware provided in the high performance 12MHz 8044 microcontroller, and a board-level integrated solution. In addition, complex, expensive, and awkward connection problems are no longer a factor because the BITBUS interconnect is a serial bus requiring a simple twisted wire pair.

Open Systems — An Answer to Obsolescence

Intel's Open Systems philosophy requires systems be open to: future VLSI, all levels of integration, third party suppliers, and special requirements. In order to facilitate this design strategy, the BITBUS interconnect was developed as a standard microcontroller interface. The same benefits realized by users of Intel's MULTIBUS architecture will be realized by users of the BITBUS architecture — the ability to exploit VLSI technology without having to pay premiums for new system design, multiple supply sources, wide product selections, and competitive prices.

The Open Systems philosophy characterizes the iDCM product line. Distributed Control Modules are compatible (open) at three levels of integration: components, boards, and systems. This multilevel approach enables OEM's to adapt to new business environments and opportunities as VLSI technology evolves.

Distributed Control Modules

The iDCM product line consists of both software and hardware products: the iRMX™ 51 Executive, iRMX 510 Support Package, iSBX™ 344 BITBUS MULTIMODULE™ Board, and the iRCB 44/10 BITBUS Remote Controller Board. All iDCM hardware products include integral firmware to implement the high-level BITBUS interfaces: message formats, command sequences, and operating system environments.

iRMX™ 51 Executive

The iRMX 51 Executive is a compact, easy to use, software tool for development and implementation of applications built on the high performance 8-bit family of 8051 microcontrollers. A pre-configured version of the Executive is included in firmware of the two iDCM hardware products. During run-time, some of the services provided by this event driven Executive are: task scheduling, interrupt handling, and message passing. Streamlined code, the simple user interface and modular design of the iRMX 51 Executive enhance system reliability.

iRMX™ 510 iDCM Support Package

The iRMX 510 iDCM Support Package provides the software development and run-time support for BITBUS systems. Also included are the software interfaces for other operating systems: iRMX 86, iRMX 286R, iRMX 88, and the iPDS ISIS. These software interfaces ease integration of a BITBUS system into MULTIBUS or iPDS environments.

iSBX™ 344 BITBUS™ MULTIMODULE™ Board

The iSBX 344 board facilitates expansion of MULTIBUS and iPDS systems via the BITBUS interconnect. This board is the iDCM hardware interface. MULTIBUS system capabilities can be expanded to include low-cost remote control using the iSBX 344 MULTIMODULE board and the iRMX 510 iDCM Support Package. Also, BITBUS system capabilities can be expanded using this board with user supplied software. The iSBX 344 board's integral firmware reduces application development time, ensures real time response, eases system integration, and lowers system cost.

iRCB 44/10 BITBUS Remote Controller Board

The iRCB 44/10 BITBUS Remote Controller module is a low-end, single-board computer with 24 lines of parallel I/O. The board has a single-high Eurocard form factor with a DIN connector for increased reliablity and integration with standard industrial packaging. One iSBX I/O Expansion connector will accommodate one of many iSBX MULTIMODULE Boards for I/O expansion. Also, sockets for repeaters are provided for extending the BITBUS interconnect beyond the length limits of one BITBUS segment. This board lowers distributed system cost via the BITBUS interconnect support and the same integral firmware provided on the iSBX 344 BITBUS Controller MULTIMODULE board.

Expanding a MULTIBUS® System with Distributed Control Modules

An example of how a MULTIBUS system can be expanded with iDCM Modules is shown in Figures 1 and 2. Figure 1 shows a basic MULTIBUS system: processor board, memory module, and I/O controller. Figure 2 illustrates the expanded system. Some advantages of the expanded system follow: The burden on the central processor has been reduced, thereby increasing overall system performance. System cost reduction is realized because the BITBUS architecture removes the necessity of adding expensive centralized systems to handle increased performance demands. Also, the BITBUS architecture enables implementation of a more efficient and flexible system that is insensitive to the addition of more nodes, or changes in node job functions.

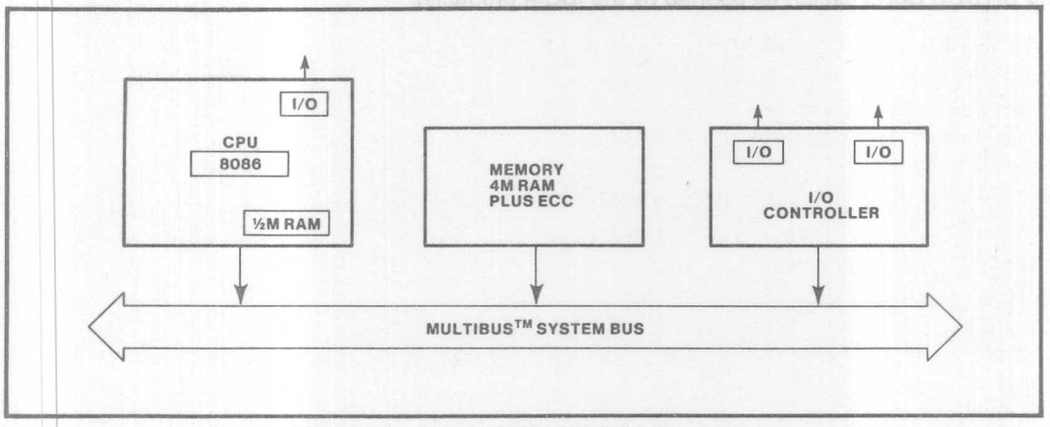


Figure 1. Basic MULTIBUS® System.

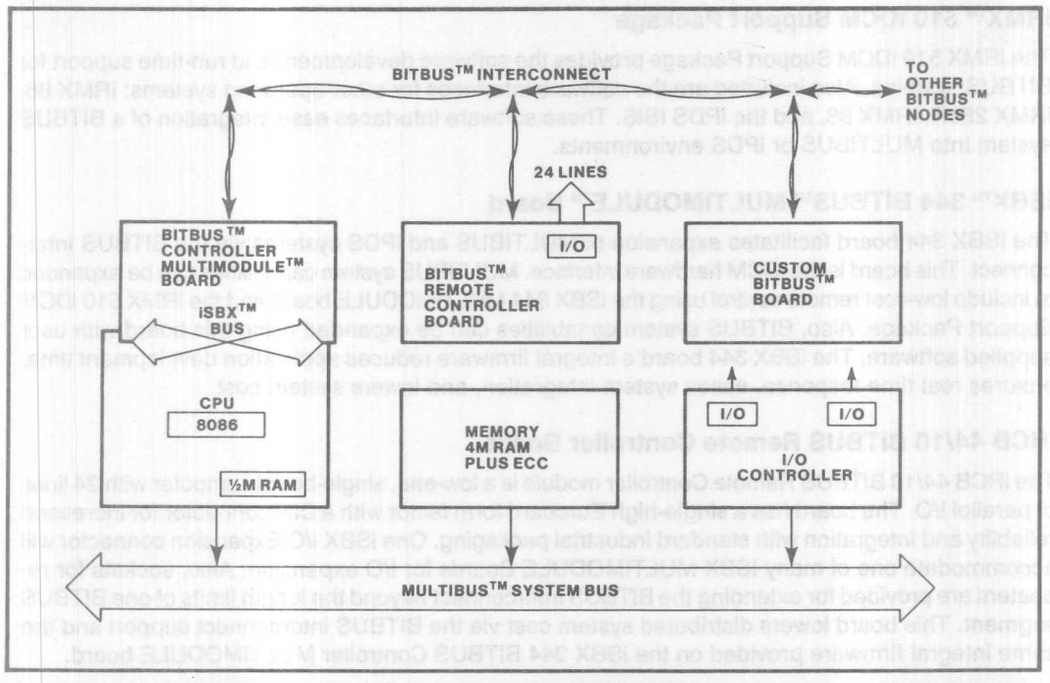


Figure 2. MULTIBUS® System Expanded with BITBUS™ Interconnect and Distributed Control Modules.

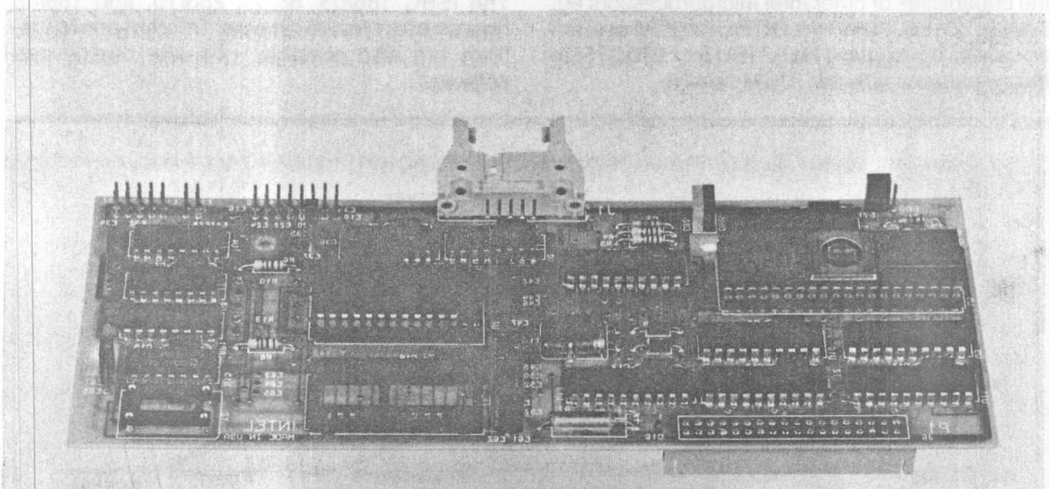
Summary

This overview has introduced the attributes and advantages of the BITBUS interconnect and Distributed Control Modules. Initial iDCM products (the iRMX 51 Executive, the iRMX 510 iDCM Support Package, iSBX 344 BITBUS Controller MULTIMODULE and iRCB 44/10 BITBUS Remote Controller boards) are intended to allow rapid assimilation of this new technology. Data sheets describing the individual iDCM products are included in the next section of this document. The final section presents the BITBUS Specification (supported by the iDCM products).

iSBX™ 344
BITBUS™ CONTROLLER MULTIMODULE™ BOARD

- **High performance 12Mhz 8044 controller**
- **Integral firmware including the iRMX™ 51 Executive optomized for real-time control applications**
- **Full BITBUS™ support**
- **2 28-pin JEDEC memory sites for user's control functions**
- **Low cost, double-wide iSBX™ BITBUS™ expansion MULTIMODULE™ board**
- **Power up diagnostics increase reliablity**

The iSBX™ 344 BITBUS™ Controller MULTIMODULE™ board is the BITBUS gateway to all Intel products that support the iSBX I/O Expansion Interface. Based on the highly integrated 8044 component (an 8 bit 8051 microcontroller and an SDLC controller on one chip) the iSBX 344 MULTIMODULE board extends the capability of other microprocessors via the BITBUS interconnect. With the other members of Intel's Distributed Control Modules (iDCM) family, the iSBX 344 MULTIMODULE board expands Intel's OEM microcomputer system capabilities to include distributed real-time control. Like all members of the iDCM family, the iSBX 344 MULTIMODULE board includes many features that make it well suited for industrial control applications such as: data acquisition and monitoring, process control, robotics, and machine control.



OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The two iDCM hardware products, the iSBX 344 MULTIMODULE board and the iRCB 44/10 BITBUS Remote Controller Board, communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

As a member of the iDCM product line the iSBX 344 MULTIMODULE board fully supports the BITBUS microcontroller interconnect. Typically, the iSBX 344 MULTIMODULE board would be part of a node (master or slave) on the BITBUS interconnect in an iDCM system. As shown in Figure 2 the iSBX 344 MULTIMODULE board plugs into any iSBC® board with an iSBX connector.

The iSBX 344 MULTIMODULE board is the hardware interface between Intel's MULTIBUS® and iPDS™ ISIS environment and the BITBUS environment. With this interface the user can harness the capabilities of other Intel microprocessors eg: 80286, 80186, 8086 in a BITBUS/iDCM system or extend an existing MULTIBUS or iPDS ISIS-based system with the iDCM family.

MULTIBUS® and iPDS™ I/O Expansion

Typically, MULTIBUS iSBC boards have a maximum of two iSBX I/O expansion connectors. These connectors facilitate addition of one or two iSBX I/O MULTIMODULE boards with varying numbers of I/O lines. The iSBX 344 MULTIMODULE board increases the number of I/O lines that can be accommodated by a MULTIBUS system by at least an order of magnitude. The iSBX 344 MULTIMODULE board extends the I/O of Intel's Personal Development System (iPDS) or other systems products in a similar manner.

Extending BITBUS™/iDCM System Processing Capability

The iSBX 344 MULTIMODULE board allows utilization of other processors in a BITBUS/iDCM system to accomodate particular application requirements. The MULTIMODULE board is compatible with any iSBX connector so that any board having a compatible connector can potentially enhance system performance. Intel's iRMX 510 iDCM Support Package provides the software interface required for a variety of iSBC boards. The iSBC 186/03, 86/30, 286/10, and 188/48 boards are a few examples. Custom configurations are also possible with user customized software.

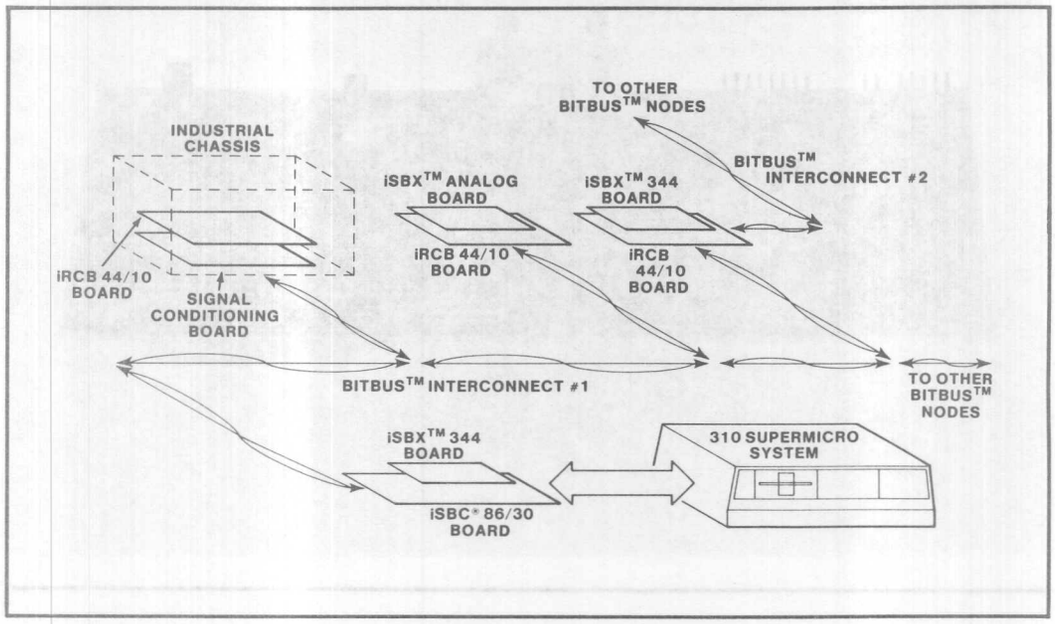


Figure 1. iDCM Operating Environment

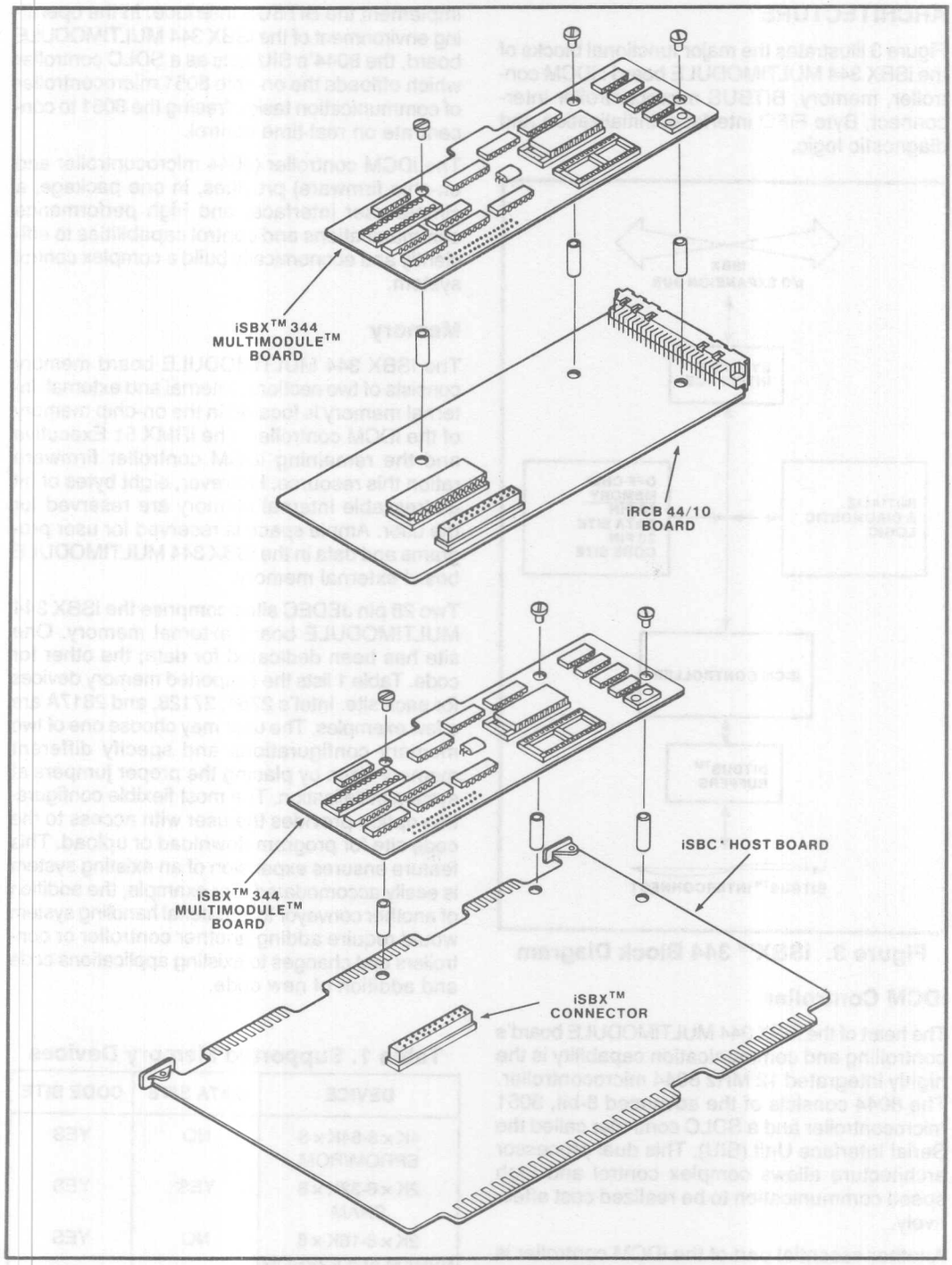


Figure 2. iSBX™ 344 Installation

ARCHITECTURE

Figure 3 illustrates the major functional blocks of the iSBX 344 MULTIMODULE board : iDCM controller, memory, BITBUS microcontroller interconnect, Byte FIFO interface, initialization and diagnostic logic.

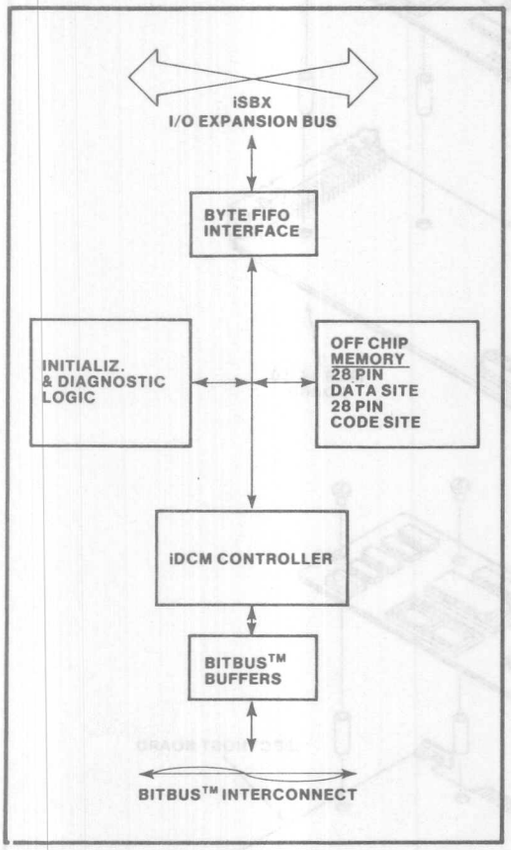


Figure 3. iSBX™ 344 Block Diagram

iDCM Controller

The heart of the iSBX 344 MULTIMODULE board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit, 8051 microcontroller and a SDLC controller called the Serial Interface Unit (SIU). This dual processor architecture allows complex control and high speed communication to be realized cost effectively.

Another essential part of the iDCM controller is the integral firmware that resides on-chip to implement the BITBUS interface. In the operating environment of the iSBX 344 MULTIMODULE board, the 8044's SIU acts as a SDLC controller which offloads the on-chip 8051 microcontroller of communication tasks; freeing the 8051 to concentrate on real-time control.

The iDCM controller (8044 microcontroller and on-chip firmware) provides, in one package, a simple user interface, and high performance communications and control capabilities to efficiently and economically build a complex control system.

Memory

The iSBX 344 MULTIMODULE board memory consists of two sections: internal and external. Internal memory is located in the on-chip memory of the iDCM controller. The iRMX 51 Executive and the remaining iDCM controller firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iSBX 344 MULTIMODULE board external memory.

Two 28 pin JEDEC sites comprise the iSBX 344 MULTIMODULE board external memory. One site has been dedicated for data; the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764, 27128, and 2817A are a few examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accomodated. For example, the addition of another conveyor to a material handling system would require adding another controller or controllers and changes to existing applications code and addition of new code.

Table 1. Supported Memory Devices

DEVICE	DATA SITE	CODE SITE
4K x 8-64K x 8 EPROM/ROM	NO	YES
2K x 8-32K x 8 SRAM	YES	YES
2K x 8-16K x 8 NVRAM and E2PROM	NO	YES

BITBUS™ Microcontroller Interconnect

The iSBX 344 MULTIMODULE board fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications. The interconnect supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission rates. Table 2 shows different combinations of modes of operations, transmission rates, and distances. The SDLC protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of this established architecture. These features contribute to BITBUS reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The BITBUS interface of the iSBX 344 MULTIMODULE board consists of a half-duplex RS 485 tranceiver and an optional clock source for the synchronous mode of operation.

Byte FIFO Interface

The Byte FIFO Interface on the iSBX 344 MULTIMODULE board implements the required hardware buffering between the iDCM controller and an extension. An extension is defined as a device attached to the iSBX I/O expansion interface on the iSBX 344 MULTIMODULE board. In an iDCM system, an example of an extension is an iSBC 86/30 board which may be considered the host board in a MULTIBUS system. When used with the software handlers in the iRMX 510 iDCM Support Package, implementation of this interface is complete.

For particular applications, the user may wish to develop a custom software interface to the extension or host board. On the iSBX 344 MULTIMODULE board side of the interface the iDCM firmware automatically accepts messages for the FIFO. No user code is required increasing the time available for application system development.

The Byte FIFO supports both byte and message transfer protocol in hardware via three register ports: data, command, and status. The extension side supports polled, interrupt, and limited DMA modes of operation (e.g. 80186 type DMA controllers).

Initalization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iSBX 344 MULTIMODULE board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an iDCM controller or external bus failure. The LEDs used for power up diagnostics are available for user diagnostics after power up as well as to further contribute to reliable operation of the system.

Initial iSBX 344 MULTIMODULE board parameters are set by positioning jumpers. The jumpers determine the BITBUS mode of operation: synchronous, self-clocked, transmission rate, and address of the iSBX module in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

The iSBX BITBUS Controller MULTIMODULE board contains resident firmware located in the iDCM Controller. The on-chip firmware consists of: a pre-configured iRMX 51 Executive for user program development; a Remote Access and Control (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to

Table 2. BITBUS™ Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Between Repeaters	Maximum # Repeaters
Synchronous	2400	30/100	28	0
Self Clocked	375	300/1000	28	2
	62.5	1200/4000	28	10

connect the BITBUS interconnect, iSBX bus, and iRMX 51 Executive tasks; and power up diagnostics.

The iRMX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iRMX 51 user interfaces. Both the executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operations transparent.

The Remote Access and Control Function is a special purpose task that allows the user to transfer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function. Power up tests provide a quick diagnostic service.

The services provided by the iSBX 344 MULTIMODULE board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution .

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the iSBX 344 MULTIMODULE board. Software development support consists of : the 8051 Software Development Package, and the iRMX 510 iDCM Support Package. The 8051 Software Development Package provides the RL 51 Linker and Relocator Program, and ASM 51. PL/M 51 is also available. The iRMX 510 Support Package includes the iDCM Controller firmware files on diskette as well as iRMX 51 libraries. Hardware tools consist of of the IN-Circuit Emulator (ICE-44), Intel's Portable Development System (iPDS), and Intellec Series II or III Development Systems.

Table 3. iRMX™ 51 Interfaces

COMMAND	DESCRIPTION
RQ SEND MESSAGE	Sends a message (a command from the BITBUS master, a response from a slave, or a simple message between tasks on the same BITBUS component) to another task.
RQ WAIT	Waits for an interrupt, and event time-out, a message, or any combination of the three.
RQ CREATE TASK	Causes a new sequence of code to be run as an iRMX 51 task with a specific function identification code and priority.
RQ DELETE TASK	Stops the specified task and removes it from all execution lists.
RQ ALLOCATE	Allocates a fixed-length buffer from the on-chip, scratch-pad RAM for general use, or, in BITBUS applications, for a BITBUS message buffer.
RQ DEALLOCATE	Returns an on-chip buffer to the system.
RQ SET INTERVAL	Set the time interval to be used as a separate event-timer for the task.
RQ ENABLE INTERRUPT	Allow external interrupts to signal the microcontroller.
RQ DISSABLE INTERRUPT	Stops all external interrupts from signaling the microcontroller.
RQ GET FUNCTION ID	Provides a list of the 8 function identification codes representing the tasks currently operating on the microcontroller.

Table 4. RAC Services

COMMAND	DESCRIPTION
READ I/O	Read external I/O location. Return result in reply message.
WRITE I/O	Write byte to external I/O location.
UPDATE I/O	Write byte to, then read byte from external I/O location. Return result in reply message.
OR I/O	OR data with contents of external I/O location. Return OR'd value.
AND I/O	AND data with contents of external I/O location. Return AND'd value.
XOR I/O	XOR data with contents of external I/O location. Return XOR'd value.
READ INTERNAL MEMORY	Read contents of internal memory location. Return result in reply message.
WRITE INTERNAL MEMORY	Write data to internal memory location.
DOWNLOAD EXTERNAL MEMORY	Write data starting at external memory location.
UPLOAD EXTERNAL MEMORY	Read data starting at external memory location. Return result in reply message.
GET FUNCTIONS	Provides a list of the 8 function identification codes representing the tasks currently operating on the microcontroller.
CREATE TASK	Causes a new sequence of code to be run as in the iRMX™ 51 interface.
DELETE TASK	Stops the specified task and removes it from all execution lists as in the iRMX™ 51 interface.
RAC PROTECT	Suspends or resumes RAC Services.
RESET DEVICE	Returns device software to original state at initialization.

NOTES:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers memory outside the microcontroller — the 28-pin sockets of the iSBX 344 module and the iRCB 44/10 board. Each RAC Access Function may refer to 1, 2, 3, 4, 5, or 6 individual I/O or memory locations in a single command.

SPECIFICATIONS

Word Size

Instruction — 8 bit
Data — 8 bit

Processor Clock 12 MHz

Instruction Execution Times
1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Capacity/Addressing

iDCM Controller — 64 K

Address Range

	Option 1	Option 2
External Memory		
Data	0000H-7FFFH	0000H-7FFFH
Code	1000H-0FFFFH	8000H-0FFEFH
Internal Memory		
Code	0000H-0FFFH	0000H-0FFFH

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible — 120 ohms or greater.

iDCM Controller (8044 + firmware) I/O Addressing as viewed from the 8044

FUNCTION	ADDRESS	READ	WRITE	BIT	COMMENTS
Data	FF00H	✓	✓		
Command	FF01H	✓	✓		Write sets command to extension — Read clears command from extension
Status					
-RFNF*	B3H	✓		✓	Also INT1 Input
-TFNE*	B2H	✓		✓	Also INT0 Input
-TCMD*	92H	✓		✓	
LED #1	90H	✓	✓	✓	
LED #2	91H	✓	✓	✓	
RDY/NE*	B4H	✓	✓	✓	
Node Address	FFFFH	✓			
Configuration	FFFEH	✓			

iSBX™ 344 MULTIMODULE™ board I/O Addressing as viewed from the iSBX™ 344 MULTIMODULE™ board

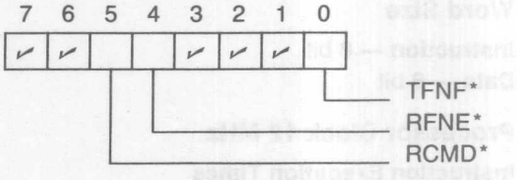
REGISTER FUNCTION	ADDRESS	COMMENTS
Data	Base	Read/Write
Command	Base + 1	Write sets command from extension Read clears command to extension
Status	Base + 2	Read Only

NOTES:
1. Base is determined by MCSO* on extension device

Interrupt/DMA Lines

SIGNAL	LOCATION	INTERFACE OPTION
RINT	MDRQ/MINT0	INT
TINT	MINT1	INT
RCMI	OPT0	INT OR DMA
RDRQ	MDRQ/MINT0	DMA
TDRQ	MINT1	DMA

Status Register Interface



Connector Options

10 Pin Plug

Flat Cable — 3M 3473-6010, TB Ansley 609-1001M, or equal
Discrete Wire — BERG 65846-007, ITT Cannon 121-7326-105, or equal

Pin Out

PIN	SIGNAL
1	+12V
2	+12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

Electrical Characteristics

Interfaces

iSBX™ I/O expansion bus — supports the standard I/O Expansion Bus Specification with compliance level D8

Memory Sites — Both code and data sites support the electrical Universal Memory Site specification

BITBUS™ Interconnect — The iSBX 344 MUL MODULE board supports the BITBUS Specific tion as follows:

Fully supported synchronous mode and s clocked mode for 375 kbits/sec and 62.5 kbi sec

The iSBX 344 MULTIMODULE board preser one standard load to the BITBUS bus

Message length of 18 bytes supported

RAC Function support as shown in Table 4

Power Requirements

.9A at +5V ± 5% iSBX™ 344 MULTIMODULE board only — memory NOT included

Physical Characteristics

Double-wide iSBX™ MULTIMODULE™ For Factor

Dimensions

Height — 10.16 mm (0.4 in) maximum co ponent height
Width — 63.5 mm (2.50 in)
Depth — 190.5 mm (7.50 in)
Weight — 113 gm (4 ounces)

Environmental Characteristics

Operating Temperature — 0°C to 55°C at 2(Linear Feet/Minute Air Velocity
Humidity — 90% non-condensing

Reference Manual (NOT Supplied)

146312 — Guide to Using the Distributed Co trol Modules

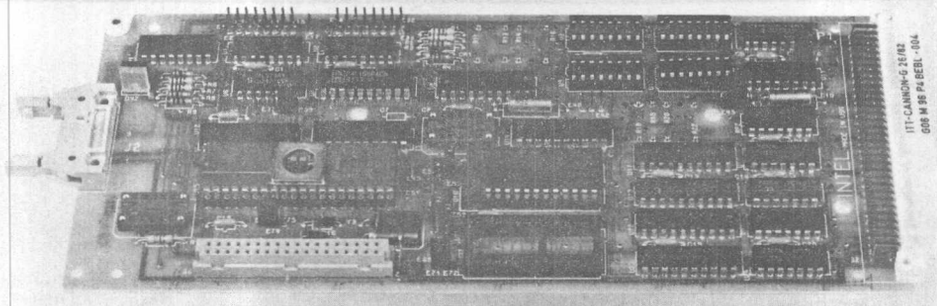
Ordering Information

Part Number	Description
iSBX 344	BITBUS Controller MULTIMODULE board

iRCB 44/10
BITBUS™ Remote Controller Board

- **High performance 12Mhz 8044 controller**
- **Integral firmware including the iRMX™ 51 Executive optomized for real-time control**
- **Full BITBUS™ support**
- **Standard industrial packaging: Eurocard, DIN connector**
- **2 28-pin JEDEC memory sites for user's control functions**
- **Low cost I/O expansion with 8-bit iSBX™ connector**
- **Programmable control/monitoring capability of 24 I/O lines**
- **Power up diagnostics increase reliablity**

The iRCB 44/10 BITBUS™ Remote Controller Board, is an intelligent real-time controller and a remote I/O expansion device. Based on the highly integrated 8044 component (an 8 bit 8051 microcontroller and an intelligent SDLC controller on one chip) the iRCB 44/10 board provides high performance control capability at low cost. Incorporating complete BITBUS support, the iRCB 44/10 board and the other members of Intel's Distributed Control Modules (iDCM) family expand Intel's OEM microcomputer system capabilities to include distributed real-time control. Like all members of the iDCM family, the iRCB 44/10 board includes many features that make it well suited for industrial control applications such as: data acquisition and monitoring, process control, robotics, and machine control.



OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The two iDCM hardware products, the iSBX 344 BITBUS controller MULTIMODULE™ board and the iRCB 44/10 BITBUS Remote Controller Board, communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

The iRCB 44/10 board can be used as an intelligent remote controller or an I/O expansion device. When performing as an intelligent controller the iRCB 44/10 board not only monitors the status of multiple process points, but it can execute varied user supplied control algorithms. When functioning as an I/O expansion device, the iRCB 44/10 board simply collects data from multiple I/O ports and transmits this information via the BITBUS or iSBX bus interface to the system controller for analysis or updating purposes.

As a member of the iDCM product line the iRCB 44/10 board fully supports the BITBUS microcontroller interconnect. Typically, the iRCB 44/10 board would be a node in a BITBUS system. The iRCB 44/10 board could be part of a master or slave node. (The BITBUS system supports a multidrop configuration: one master, many slaves.)

ARCHITECTURE

Figure 2 illustrates the major functional blocks of the iRCB 44/10 board: iDCM controller, memory, BITBUS microcontroller interconnect, parallel I/O, iSBX expansion, initialization and diagnostic logic.

iDCM Controller

The heart of the iRCB 44/10 board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit 8051 microcontroller and a SDLC controller called the Serial Interface Unit (SIU). This dual processor architecture allows complex control and high speed communication functions to be realized cost effectively.

Another essential part of the iDCM controller is the integral firmware that resides on-chip to implement the BITBUS interface. In the operating environment of the iRCB 44/10 board, the 8044's SIU acts as a SDLC controller which offloads the on-chip 8051 microcontroller of communication tasks; freeing the 8051 to concentrate on real-time control.

The iDCM controller (8044 microcontroller and on-chip firmware) provides, in one package, a simple user interface, and high performance communications and control capabilities to efficiently and economically build a complex control system.

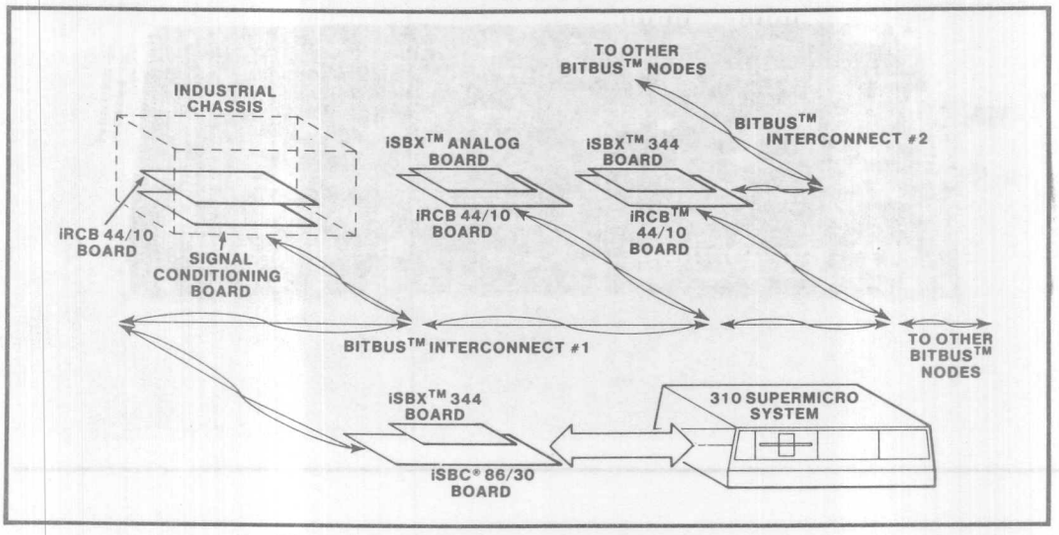


Figure 1. iDCM Operating Environment

Memory

The iRCB 44/10 board memory consists of two sections: internal and external. Internal memory is located in the on-chip memory of the iDCM controller. The iRMX 51 Executive and the remaining iDCM controller firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iRCB 44/10 board external memory.

Two 28 pin JEDEC sites comprise the iRCB 44/10 board external memory. One site has been dedicated for data, the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764, 27128, and 2817A are a few examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accomodated.

Table 1. Supported Memory Devices

DEVICE	DATA SITE	CODE SITE
4K x 8-64K x 8 EPROM/ROM	NO	YES
2K x 8-32K x 8 SRAM	YES	YES
2K x 8-16K x 8 NVRAM and E2PROM	NO	YES

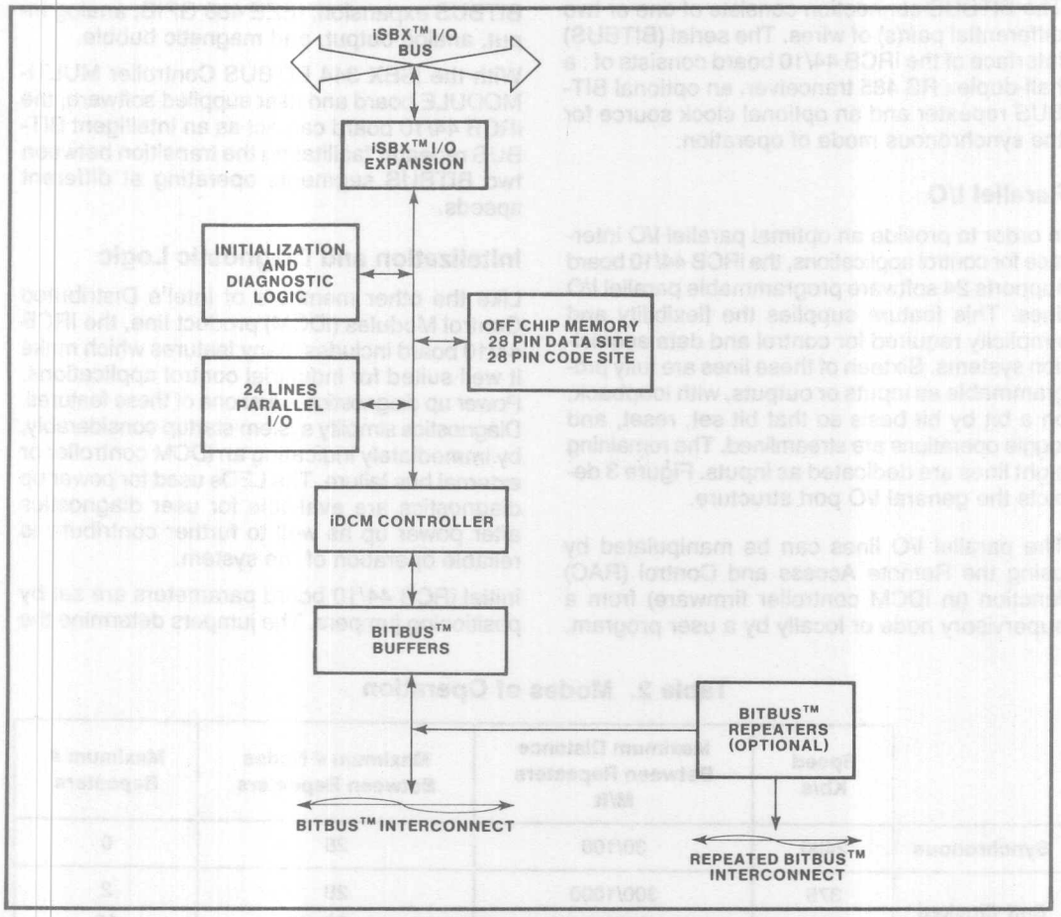


Figure 2. iRCB™ 44/10 Block Diagram

BITBUS™ Microcontroller Interconnect

The iRCB 44/10 board serial interface fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications. The bus supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission speeds. Table 2 shows the different combinations of modes of operation, transmission speeds, and distances. The SDLC protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of the BITBUS architecture. These features contribute to BITBUS system reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The serial (BITBUS) interface of the iRCB 44/10 board consists of : a half-duplex RS 485 tranceiver, an optional BITBUS repeater and an optional clock source for the synchronous mode of operation.

Parallel I/O

In order to provide an optimal parallel I/O interface for control applications, the iRCB 44/10 board supports 24 software programmable parallel I/O lines. This feature supplies the flexibility and simplicity required for control and data acquisition systems. Sixteen of these lines are fully programmable as inputs or outputs, with loopback, on a bit by bit basis so that bit set, reset, and toggle operations are streamlined. The remaining eight lines are dedicated as inputs. Figure 3 depicts the general I/O port structure.

The parallel I/O lines can be manipulated by using the Remote Access and Control (RAC) function (in iDCM controller firmware) from a supervisory node or locally by a user program. The user program can also access the RAC function or directly operate the I/O lines. Input, output, mixed — input and output, and bit operations are possible simply by reading or writing a particular port.

iSBX™ Expansion

One iSBX I/O expansion connector is provided on the iRCB 44/10 board. This connector can be used to extend the I/O capability of the board. In addition to specialized and custom designed iSBX boards, a full line of compatible high speed, 8-bit expansion MULTIMODULE boards, both single and double wide, are available from Intel. The only incompatible modules are those that require the MWAIT* signal or DMA operation. A few of Intel's iRCB 44/10 board compatible iSBX MULTIMODULE boards include: parallel I/O, serial I/O, BITBUS expansion, IEEE 488 GPIB, analog input, analog output, and magnetic bubble.

With the iSBX 344 BITBUS Controller MULTIMODULE board and user supplied software, the iRCB 44/10 board can act as an intelligent BITBUS repeater facilitating the transition between two BITBUS segments operating at different speeds.

Initalization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iRCB 44/10 board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an iDCM controller or external bus failure. The LEDs used for power up diagnostics are available for user diagnostics after power up as well to further contribute to reliable operation of the system.

Initial iRCB 44/10 board parameters are set by positioning jumpers. The jumpers determine the

Table 2. Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Between Repeaters	Maximum # Repeaters
Synchronous	2400	30/100	28	0
Self Clocked	375	300/1000	28	2
	62.5	1200/4000	28	10

BITBUS mode of operation: synchronous, self clocked, transmission speed, and address of the iRCB 44/10 board in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

The iRCB 44/10 board contains resident firmware located in the iDCM Controller. The on-chip firmware consists of: a pre-configured iRMX 51 Executive for user program development; a Remote Access and Control (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to connect the BITBUS interconnect, iSBX bus, and iRMX 51 tasks; and power up diagnostics.

The iRMX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iRMX 51 user interfaces. Both the Executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operations transparent.

The Remote Access and Control Function is a special purpose task that allows the user to trans-

Table 3. iRMX™ 51 Interfaces

COMMAND	DESCRIPTION
RQ SEND MESSAGE	Sends a message (a command from the BITBUS master, a response from a slave, or a simple message between tasks on the same BITBUS component) to another task.
RQ WAIT	Waits for an interrupt, and event time-out, a message, or any combination of the three.
RQ CREATE TASK	Causes a new sequence of code to be run as an iRMX 51 task with a specific function identification code and priority.
RQ DELETE TASK	Stops the specified task and removes it from all execution lists.
RQ ALLOCATE	Allocates a fixed-length buffer from the on-chip, scratch-pad RAM for general use, or, in BITBUS applications, for a BITBUS message buffer.
RQ DEALLOCATE	Returns an on-chip buffer to the system.
RQ SET INTERVAL	Set the time interval to be used as a separate event-timer for the task.
RQ ENABLE INTERRUPT	Allow external interrupts to signal the microcontroller.
RQ DISSABLE INTERRUPT	Stops all external interrupts from signalling the microcontroller.
RQ GET FUNCTION ID	Provides a list of the 8 function identification codes representing the tasks currently operating on the microcontroller.

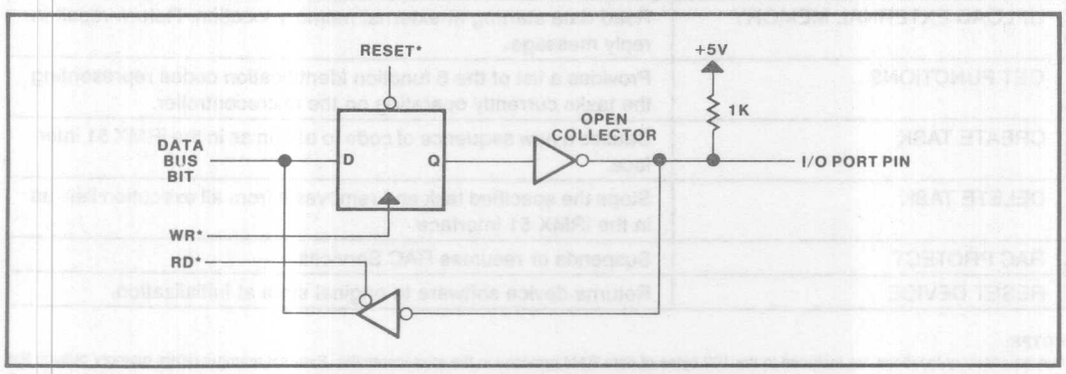


Figure 3. I/O Port Structure

fer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function. Power up tests provide a quick diagnostic service.

The services provided by the iRCB 44/10 board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution.

INDUSTRIAL PACKAGING

The iRCB 44/10 form factor is a single high, 220mm deep Eurocard as shown in Figure 4. The Eurocard form factor supports most standard industrial packaging schemes as well as Intel's MULTIBUS® II packaging scheme. The Eurocard form factor specifies reliable DIN connectors. A standard 64 pin connector is included on the iRCB 44/10 board.

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the iRCB 44/10 board. Software development support consists of: the 8051 Software Development Package, and the iRMX 510 iDCM Support Package. The 8051 Software Development Package provides the RL 51 Linker and Relocator Program, and ASM 51. PL/M 51 is also available. The iRMX 510 Support Package includes the iDCM Controller firmware files on diskette, as well as iRMX 51 libraries. Hardware tools consist of of the In-Circuit Emulator (ICE-44), Intel's Portable Development System (iPDS), and Intellec Series II or III Development Systems.

Table 4. RAC Services

COMMAND	DESCRIPTION
READ I/O	Read external I/O location. Return result in reply message.
WRITE I/O	Write byte to external I/O location.
UPDATE I/O	Write byte to, then read byte from external I/O location. Return result in reply message.
OR I/O	OR data with contents of external I/O location. Return OR'd value.
AND I/O	AND data with contents of external I/O location. Return AND'd value.
XOR I/O	XOR data with contents of external I/O location. Return XOR'd value.
READ INTERNAL MEMORY	Read contents of internal memory location. Return result in reply message.
WRITE INTERNAL MEMORY	Write data to internal memory location.
DOWNLOAD EXTERNAL MEMORY	Write data starting at external memory location.
UPLOAD EXTERNAL MEMORY	Read data starting at external memory location. Return result in reply message.
GET FUNCTIONS	Provides a list of the 8 function identification codes representing the tasks currently operating on the microcontroller.
CREATE TASK	Causes a new sequence of code to be run as in the iRMX 51 interface.
DELETE TASK	Stops the specified task and removes it from all execution lists as in the iRMX 51 interface.
RAC PROTECT	Suspends or resumes RAC Services.
RESET DEVICE	Returns device software to original state at initialization.

NOTES:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers memory outside the microcontroller — the 28-pin sockets of the iSBX 344 module and the iRCB 44/10 board. Each RAC Access Function may refer to 1, 2, 3, 4, 5, or 6 individual I/O or memory locations in a single command.

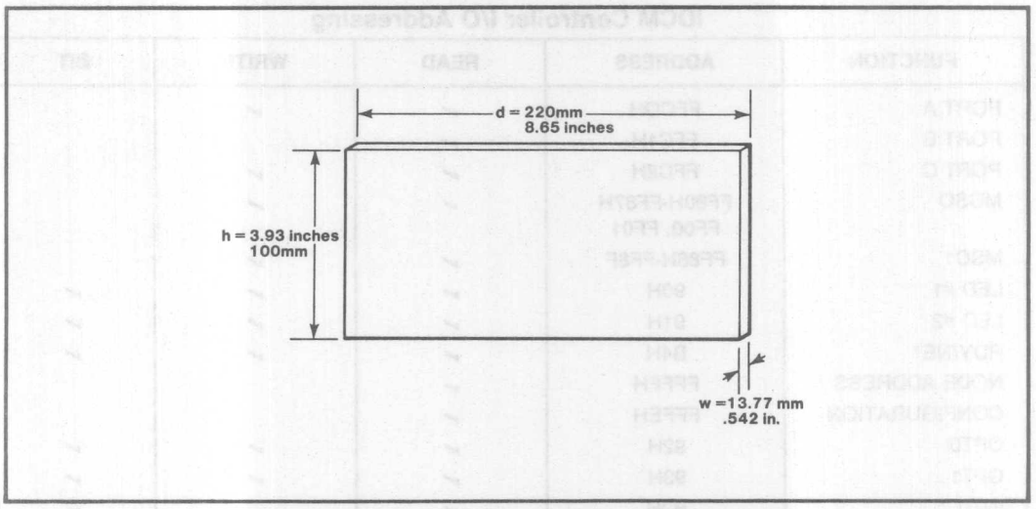


Figure 4. Eurocard Single High Form Factor

SPECIFICATIONS

Word Size

Instruction — 8 bit
Data — 8 bit

Processor Clock 12 MHz

Instruction Execution Times
1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Capacity/Addressing

iDCM Controller — 64 K

Address Range

	Option 1	Option 2
External Memory		
Data	0000H-7FFFH	0000H-7FFFH
Code	1000H-0FFFFH	8000H-0FFEFH
Internal Memory		
Code	0000H-0FFFH	0000H-0FFFH

I/O Capacity

iSBX MULTIMODULE™ board — one single or doublewide not requiring MWAIT* or DMA
24 Digital Lines Programmable Parallel I/O

Interrupt Sources

Two external — iSBX I/O Expansion bus sources or other sources.
BITBUS Microcontroller Interconnect.

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible — 120 ohms or greater.

Repeaters

Sockets provided on board — Devices 75174 and 75175

Connector Options

10 Pin Plug

Flat Cable — 3M 3473-6010, TB Ansley 609-1001M, or equal
Discrete Wire — BERG 65846-007, ITT Cannon 121-7326-105, or equal

DIN Connector Plug

Flat Cable — GW Elco 00-8259-096-84-124, Robinson Nugent RNE-IDC64C-TG30, or equal
Discrete Wire — ITT Cannon G06 M96 P3 BDBL-004 GW Elco 60 8257 3017, or equal

iDCM Controller I/O Addressing

FUNCTION	ADDRESS	READ	WRITE	BIT
PORT A	FFCOH	✓	✓	
PORT B	FFC1H	✓		
PORT C	FFC2H	✓	✓	
MCSO	FF80H-FF87H FF00, FF01	✓	✓	
MSC1	FF88H-FF8F	✓	✓	
LED #1	90H	✓	✓	✓
LED #2	91H	✓	✓	✓
RDY/NE*	B4H	✓	✓	✓
NODE ADDRESS	FFFFH	✓		
CONFIGURATION	FFFEH	✓		
OPT0	92H	✓	✓	✓
OPT1	93H	✓	✓	✓
INT0	B2H	✓		✓
INT1	B3H	✓		✓

10 PIN REPEATER CONNECTOR PIN OUT

PIN	SIGNAL
1	+ 12V
2	+ 12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

iRCB 44/10 Pin Out

DIN Connector

DIN PIN #	PIN & SOCKET PIN #	FUNCTION	DIN PIN #	PIN & SOCKET PIN #	FUNCTION
1a		GND	1c		GND
2a		+5V	2c		+5V
3a		DATA	3c		DATA*
4a		DLCK/RT	4c		DLCK*/RTS*
5a	1	EXTINT	5c		RGND
6a	3	PB7	6c	2	GND
7a	5	PB6	7c	4	GND
8a	7	PB5	8c	6	GND
9a	9	PB4	9c	8	GND
10a	11	PB3	10c	10	GND
11a	13	PB2	11c	12	GND
12a	15	PB1	12c	14	GND
13a	17	PB0	13c	16	GND
14a	19	PC3	14c	18	GND
15a	21	PC2	15c	20	GND
16a	23	PC1	16c	22	GND
17a	25	PC0	17c	24	GND
18a	27	PC4	18c	26	GND
19a	29	PC5	19c	28	GND
20a	31	PC6	20c	30	GND
21a	33	PC7	21c	32	GND
22a	35	PA7	22c	34	GND
23a	37	PA6	23c	36	GND
24a	39	PA5	24c	38	GND
25a	41	PA4	25c	40	GND
26a	43	PA3	26c	42	GND
27a	45	PA2	27c	44	GND
28a	47	PA1	28c	46	GND
29a	49	PA0	29c	48	GND
30a		+12V	30c		−12V
31a		+5V	31c		+5V
32a		GND	32c		GND

Electrical Characteristics

Interfaces

iSBX I/O expansion bus — supports the standard I/O Expansion Bus Specification with compliance level D8/8F

Memory Sites — Both code and data sites support the electrical Universal Memory Site specification

BITBUS™ Interconnect — The iRCB 44/10 Remote Controller Board supports the BITBUS Specification as follows:

Fully supported synchronous mode and self clocked mode for 375 kbits/sec and 62.5 kbits/sec

The iRCB 44/10 Remote Controller Board presents one standard load to the BITBUS bus without repeaters, with repeaters two standard loads

Message length of 18 bytes supported

RAC Function support as shown in Table 4

Parallel I/O — See the Table 5 for Electrical Specifications of the interface.

Power Requirements

.9A at +5V ± 5% iRCB 44/10 board only — memory, repeater, or iSBX board NOT included

Physical Characteristics

Single high, 220mm deep Eurocard Form Factor

Dimensions

Width — 13.77 mm (.542 in) maximum component height

Height — 100 mm (3.93 in)

Depth — 220 mm (8.65 in)

Weight — 169 gm (6 ounces)

Environmental Characteristics

Operating Temperature — 0°C to 55°C at 200 Linear Feet/Minute Air Velocity

Humidity — 90% non-condensing

Reference Manual (NOT Supplied)

146312 — Guide to Using the Distributed Control Modules

Table 5. Parallel I/O Electrical Specification

PARAMETER	CONDITION	MIN	MAX	UNITS
V_{OL}	$I_{OL} = 16$ mA		0.5	V
V_{OH}	$I_{OH} = -2$ mA	2.4		V
V_{IH}		2.0	7.0	V
V_{IL}		−1.0	0.8	V
I_{IL}	$V_{IL} = 0.5V$		6.0	mA
I_{IH}	V_{IH} = logic high		.0	mA
I_I	$V_{IH} = 7V$		−2.2	mA

Ordering Information

Part Number	Description
iRCB 44/10	BITBUS Remote Controller board

intel

iRMX™ 51
REAL-TIME MULTITASKING EXECUTIVE

- **Software tool for family of 8051 microcontroller based applications**
- **Real-time, multitasking executive**
- **Supports remote task communication**
- **Small — 2.2K Bytes**
- **Reliable**
- **Simple user interface**
- **Compatible with BITBUS™/Distributed Control Modules (iDCM) product line: iSBX™ 344 & iRCB 44/10 boards**

The iRMX™ 51 Executive is a compact, easy to use, software tool for development and implementation of applications built on the high performance 8-bit family of 8051 microcontrollers. A few members of this expansive family are the 8051, 8044, and 8052 microcontrollers. Like the 8051 family, the iRMX 51 Executive incorporates many features that make it exceptionally well suited for real-time control applications requiring manipulation and scheduling of more than one job, and fast response to external stimuli.

The 8051 microcontroller family is the family of choice for applications such as: data acquisition and monitoring, process control, robotics, and machine control. Using the iRMX 51 Executive for a foundation can significantly reduce applications development time. Also, the iRMX 51 Executive fully supports Intel's BITBUS™ microcontroller interconnect expressly designed for reliable high performance real-time control.

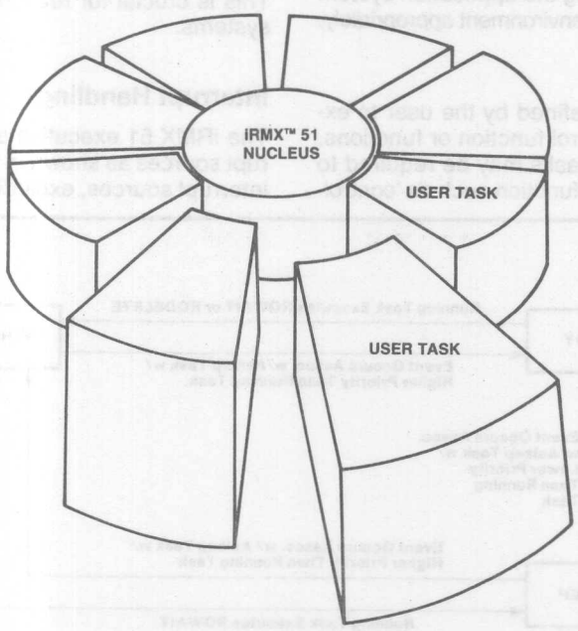


Figure 1. Structure Diagram

ARCHITECTURE

Real-time and Multitasking

Real-time control applications must be responsive to the external environment and typically involve the execution of more than one function (task or set of tasks) in response to different external stimuli. Control of an industrial drying process is an example. This process could require monitoring of multiple temperatures and humidity; control of fans, heaters, and motors that must respond accordingly to a variety of inputs. The iRMX 51 Executive fully supports applications requiring response to stimuli as they occur ie. in real-time. This real-time response is supported for multiple tasks often needed to implement a control application.

Some of the facilities precisely tailored for development and implementation of real-time control application systems provided by the iRMX 51 Executive are: task management, interrupt handling, message passing, and when intergrated with communications support, message passing with different microcontrollers. Also, the iRMX 51 Executive is driven by events: interrupts, timers, and messages ensuring the application system always responds to the environment appropriately.

Task Management

A task is a program defined by the user to execute a particular control function or functions. Multiple programs or tasks may be required to implement a particular function such as 'controlling Heater 1'. The iRMX 51 Executive recognizes three different task states as one of the mechanisms to accomplish scheduling of up to eight tasks. Figure 2 illustrates the different task states and their relationship to one another.

The scheduling of tasks is priority based. The user can prioritize tasks to reflect their relative importance within the overall control scheme. For instance, if Heater 1 must go off line prior to Heater 2 then the task associated with Heater 1 shutdown could be assigned a higher priority ensuring the correct shutdown sequence. The RQ WAIT system call is also a scheduling tool. In this example the task implementing Heater 2 shutdown could include an instruction to wait for completion of the task that implements Heater 1 shutdown.

The iRMX 51 Executive allows for PREEMPTION of a task that is currently being executed. This means that if some external event occurs such as a catastrophic failure of Heater 1, a higher priority task associated with the interrupt, message, or timeout resulting from the failure will preempt the running task. Preemption ensures the emergency will be responded to immediately. This is crucial for real-time control application systems.

Interrupt Handling

The iRMX 51 executive supports sixteen interrupt sources as shown in Table 1. Four of these interrupt sources, excluding timer 0, can be as-

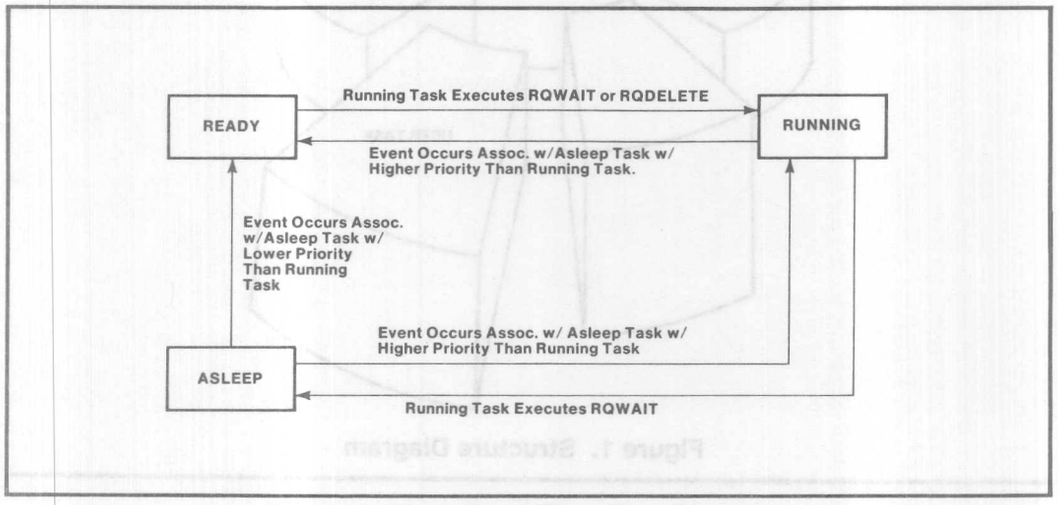


Figure 2. Task State Transition Diagram

signed to a task. When one of the interrupts occurs the task associated with it becomes a running task (if it were the highest priority task in a ready state). In this way, the iRMX 51 Executive responds to a number of internal and external stimuli including time intervals designated by the user.

Table 1. iRMX™ 51 Interrupt Sources

INTERRUPT SOURCE	INTERRUPT NUMBER
External Request 0	00H
Timer 0	01H
External Request 1	02H
Timer 1	03H
Internal Serial Port 1	04H
Reserved	05H
Reserved	06H
Reserved	07H
Reserved	08H
Reserved	09H
Reserved	0AH
Reserved	0BH
Reserved	0CH
Reserved	0DH
Reserved	0EH
Reserved	0FH

Message Passing

The iRMX 51 Executive allows tasks to interface with one another via a simple message passing facility. This message passing facility can be extended to different processors when communications support is integrated within a BITBUS/iDCM system, for example. This facility provides the user with the ability to link different functions or tasks. Linkage between tasks/functions is typically required to support development of complex control applications with multiple sensors (inputs variables) and drivers (output variables). For instance, the industrial drying process might require a dozen temperature inputs, six moisture readings, and control of: three fans, two conveyor motors, a dryer motor, and a pneumatic conveyor. The data gathered from both the temperature and humidity sensors could be processed. Two tasks might be required to gather the data and process it. One task could perform a part of the analysis, then include a pointer to the next task to complete the next part of the analysis . The tasks could continue to move between one another.

REMOTE TASK COMMUNICATION

The iRMX 51 Executive system calls can support communication to tasks on remote controllers. This feature makes the iRMX 51 Executive ideal for applications using distributed architectures. Providing communication support saves significant application development time and allows for more effective use of this time. Intel's iDCM product line combines hardware and software to provide this function.

In an iDCM system, communication between nodes occurs via the BITBUS microcontroller interconnect. The BITBUS microcontroller interconnect is a high performance serial control bus specifically intended for use in applications built on distributed architectures. The iRMX 51 Executive provides BITBUS support.

BITBUS™/iDCM COMPATIBLE

A pre-configured version of the iRMX 51 Executive implements the BITBUS message format and provides all iRMX 51 facilities mentioned previously: task management, interrupt handling, and message passing. This version of the Executive is supplied in firmware on the iDCM Controller with the iDCM hardware products: the iSBX 344 BITBUS Controller MULTIMODULE and the iRCB 44/10 BITBUS Remote Controller boards. It is also supplied on diskette as part of the iRMX 510 iDCM Support Package to ease development of BITBUS systems.

SIMPLE USER INTERFACE

The iRMX 51 Executive's capabilities are utilized through system calls. These interfaces have been defined for ease of use and simplicity. Table 2 includes a listing of these interfaces and their functions. Note tasks may be created at system initialization or run-time using the CREATE TASK call.

Functions such as GET FUNCTION IDS, ALLOCATE/DEALLOCATE BUFFER, and SEND MESSAGE (Messages in the iRMX 51 Executive have a maximum size of 64 bytes.), support communication for distributed architectures. Architectures that define multiple remote stations requiring intelligent and dumb I/O manipulation. The remaining

Table 2. iRMX™ 51 System Interfaces

COMMAND	DESCRIPTION
RQ SEND MESSAGE	Sends a message (a command from the BITBUS master, a response from a slave, or a simple message between tasks on the same BITBUS component) to another task.
RQ WAIT	Waits for an interrupt, an event time-out, a message, or any combination of the three.
RQ CREATE TASK	Causes a new sequence of code to be run as an iRMX 51 task with a specific function identification code and priority.
RQ DELETE TASK	Stops the specified task and removes it from all execution lists.
RQ ALLOCATE	Allocates a fixed-length buffer from the on-chip, scratch-pad RAM for general use, or, in BITBUS applications, for a BITBUS message buffer.
RQ DEALLOCATE	Returns an on-chip buffer to the system.
RQ SET INTERVAL	Set the time interval to be used as a separate event-timer for the task.
RQ ENABLE INTERRUPT	Allow external interrupts to signal the microcontroller.
RQ DISSABLE INTERRUPT	Stops all external interrupts from signaling the microcontroller.
RQ GET FUNCTION ID	Provides a list of the 8 function identification codes representing the tasks currently operating on the microcontroller.

interfaces allow the user to specify the system's response to the external environment — a must for real-time control.

Another feature that eases application development is automatic register bank allocation. The Executive will assign tasks to register banks automatically unless a specific request is made. The iRMX 51 Executive keeps track of the register assignments allowing the user to concentrate on other activities.

The user configures an iRMX 51 system simply by: specifying the initial set of task descriptors and configuration values, and linking the system via the RL 51 Linker and Locator Program with user programs. The nature of the task descriptors allows the user to develop programs, locate them in off-chip ROM, and access them without writing additional code. Programs may be written in ASM 51 or PL/M 51. (Intel's 8051 Software Development Package contains both ASM 51 and RL 51. The iRMX 51 Executive supplies the configuration file and macro defining initial task descriptors.) Figure 3 shows the relationships that exist in the system generation process.

RELIABILE

Real-time control applications require reliability. The nucleus requires about 2K bytes of code space, 40 bytes on-chip RAM, & 218 bytes external RAM. Streamlined code increases performance and reliability, and flexibility is not sacrificed as code may be added to either on-chip or external memory.

The iRMX 51 architecture and simple user interface further enhance reliablity and lower cost. For example, the straightforward structure of the user interfaces, and the transparent nature of the scheduling process contribute to reliability of the overall system by minimizing programming effort. Also, modularity increases reliability of the system and lowers cost by allowing user tasks to be refined independent of the system. In this way, errors are identified earlier and can be easily corrected in each isolated module.

In addition, users can assign tasks a Function ID that allows tracking of the tasks associated with a particular control/monitoring function. This feature reduces maintenance and trouble shooting time thus increasing system run time and decreasing cost.

OPERATING ENVIRONMENT

The iRMX 51 Executive supports applications development based on any member of the high performance 8051 family of microcontrollers. The Executive is available on diskette with user linkable libraries or in the Distributed Control Modules (iDCM) controller preconfigured in on-

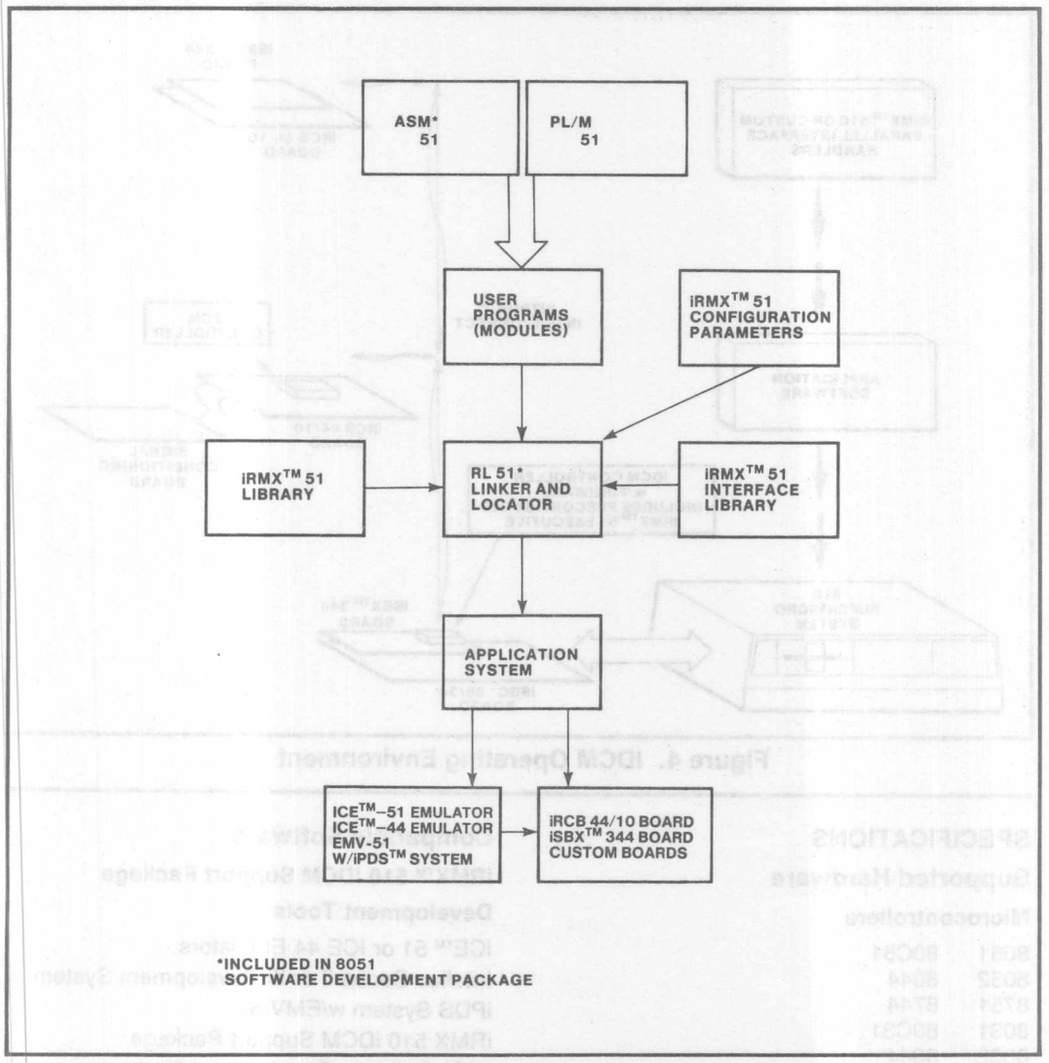


Figure 3. System Generation Process

chip ROM. (The iDCM controller is an 8044 component that consists of an 8051 microcontroller and SDLC controller on one chip with integral firmware.)

When in the iDCM environment (Figure 4), the iRMX 51 Executive can communicate with iRMX based systems like the System 286/310 or ISIS based systems like the Intel Portable Development System (iPDS) by using the iRMX 510 iDCM Support Package.

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the 8051 family of microcontrollers. This environment encompasses iDCM system (BITBUS based) applications also. Software development support consists of: the 8051 Software Development Package, and the iRMX 510 iDCM Support Package. Hardware tools consist of a variety of In Circuit Emulators (ICE), Intel's Portable Development System (iPDS) with EMV-51, and Intellec® Series II or III Development Systems.

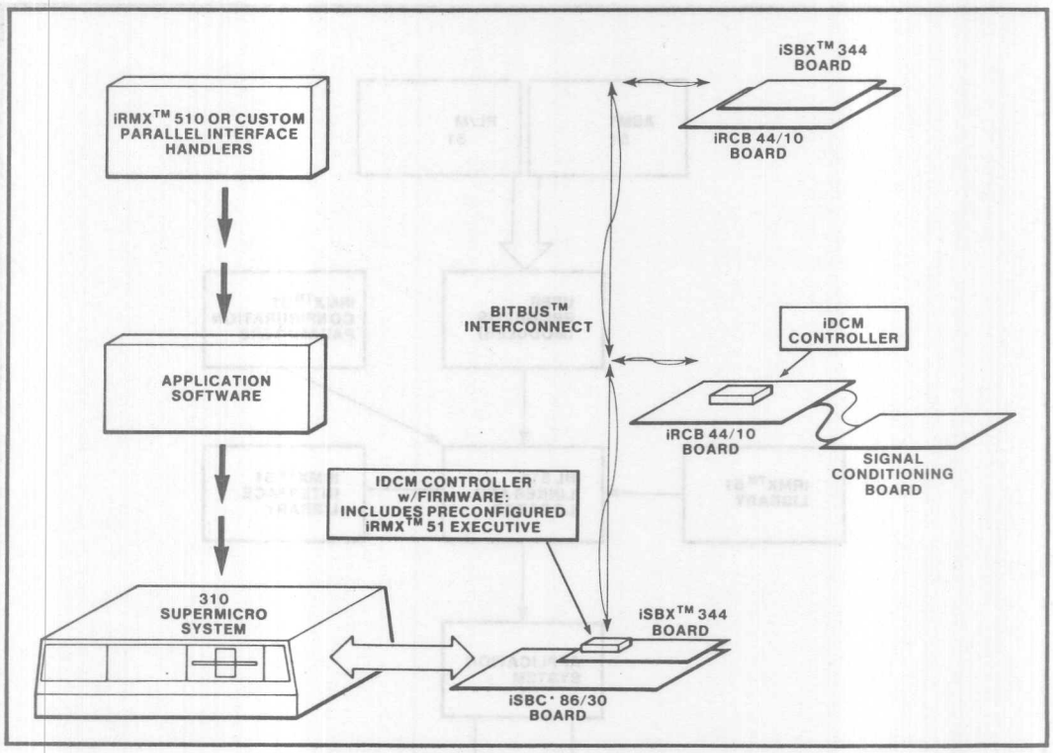


Figure 4. iDCM Operating Environment

SPECIFICATIONS

Supported Hardware

Microcontrollers

8051	80C51
8052	8044
8751	8744
8031	80C31
8032	8344

iDCM Product Line

iSBX 344 MULTIMODULE Board
iRCB 44/10 Remote Controller Board

Compatible Software

iRMX™ 510 iDCM Support Package

Development Tools

ICE™ 51 or ICE 44 Emulators
Intellec Series II or III Development System
iPDS System w/EMV-51
iRMX 510 iDCM Support Package
8051 Software Development Package

Reference Manual (Supplied)

146312-001 — Guide to Using the Distributed Control Modules

Ordering Information

Part Number	Description
iRMX 51	Executive for 8051 Family of Microcontrollers with Reference Manual. A, B, and F Media Formats Supplied

iRMX™ 510
iDCM SUPPORT PACKAGE

- **Low cost remote communication/ control expansion for MULTIBUS® based systems**
- **Extends functionality of BITBUS™/iDCM systems**
- **Software development support for BITBUS™/iDCM products: iSBX™ 344 and iRCB 44/10 boards**
- **Simple software interface for iRMX™ 86, 286, 88, and iPDS™ ISIS operating system compatibility**

The iRMX™ 510 iDCM Support Package contains the necessary software tools to interface MULTIBUS®, and iPDS™ ISIS systems to BITBUS™ systems in both a development environment and during run-time. With other members of the Distributed Control Modules family, the iRMX 510 iDCM Support Package expands Intel's OEM Microcomputer Systems capabilities to include distributed real-time control.

The iRMX 510 Package software interface handlers and the iSBX™ 344 BITBUS Controller MULTIMODULE™ board extend the capabilities of other microprocessors such as the 8086, 80186, or 80286 in iDCM, MULTIBUS, or iPDS systems. Support of iRMX 51 applications is provided via the iRMX 51 libraries incorporated in the iRMX 510 Support Package. Also, the Support Package completes the development environment for BITBUS/iDCM products : iSBX 344 and iRCB 44/10 boards. When used with an ICE-44 Emulator the iDCM controller is accurately simulated resulting in a highly effective product development effort.



MULTIBUS®, iPDS,™ and iDCM SYSTEM EXPANSION

The iRMX 510 Support Package provides the software interface between Intel's MULTIBUS and iPDS environment, and the BITBUS environment. With Intel's Distributed Control Modules hardware interface, the iSBX 344 MULTIMODULE board, this capability enables the user to expand the existing functionality of an iRMX-based SYSTEM 310, for example, to include control and monitoring of a material handling operation. Intel's Personal Development System (iPDS) can be used as a central supervisory station for data acquistion in a laboratory or for program development. The iRMX 510 iDCM Support Package provides a general purpose interface. For custom applications, users may wish to develop a custom interface.

OPERATING ENVIRONMENT

The iRMX 510 Support Package is supplied on diskettes formated for iRMX, Intellec® Series II or III and iPDS ISIS development systems. Application programs or tasks residing on an extension in the iDCM environment may use the iRMX 510 interface. (Application programs or tasks are written in iRMX 88, 86 or ISIS compatible code.) Some examples of extensions in an iDCM system are the iSBC 86/05, 88/25, 186/03 boards and the iPDS system. Figure 2 shows how the iRMX 510 interface is integrated into an iDCM system.

For iRMX 86, 88, or 286R-based systems, configuration of the iRMX 510 interface requires two steps: configuring the interface to the hardware and then the supporting executive. Hardware configuration requires creating a file of configuration parameters, compiling it, and linking the result with the application program. When using the iRMX 510 Package with the iPDS ISIS system, hardware configuration is not required.

ARCHITECTURE

The major functional blocks of the iRMX 510 Support Package are: iRMX 86, 286R, 88 and iPDS ISIS parallel interface handlers, iDCM Controller firmware files, and iRMX 51 include files.

Simple Parallel Interface Handlers

The iRMX 510 Support Package includes parallel interface handlers for systems using the iRMX 86 or 286R Operating System, the iRMX 88 Executive, or Intel's Personal Development System ISIS Operating System. These software handlers pass iRMX 51 messages to and from the iSBX 344 parallel interface (Byte FIFO). In iRMX 86, 286R or 88 — based systems , the interface executes as two tasks: one to transmit, the other to receive the message. In iPDS systems the interface is a procedural call: DCM TRANSMIT, DCM RECIEVE, or DCM STATUS CHECK. In both cases the handlers are straightforward and easy to use. Figure 1 illustrates transmission of a message in an iRMX-based system.

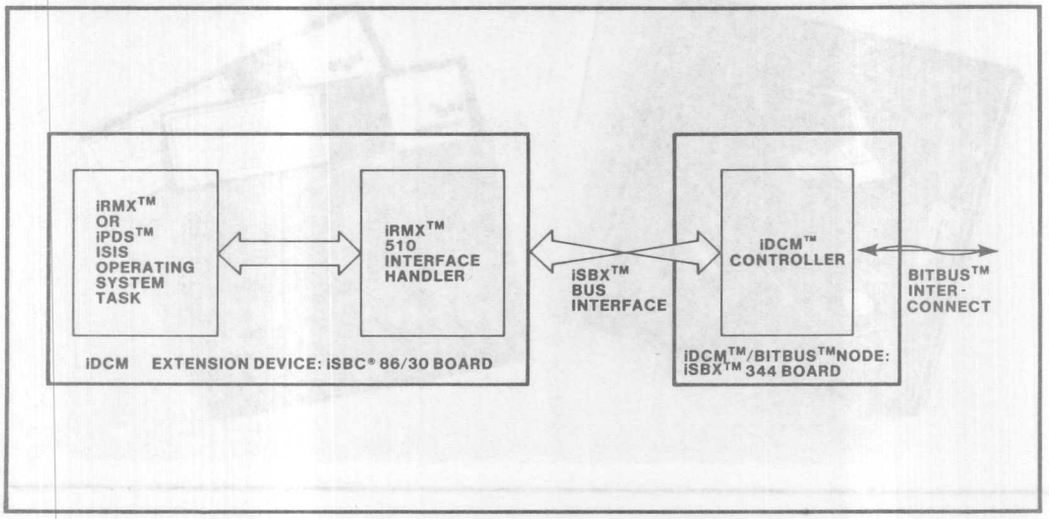


Figure 1. Message Transfer to an iDCM System

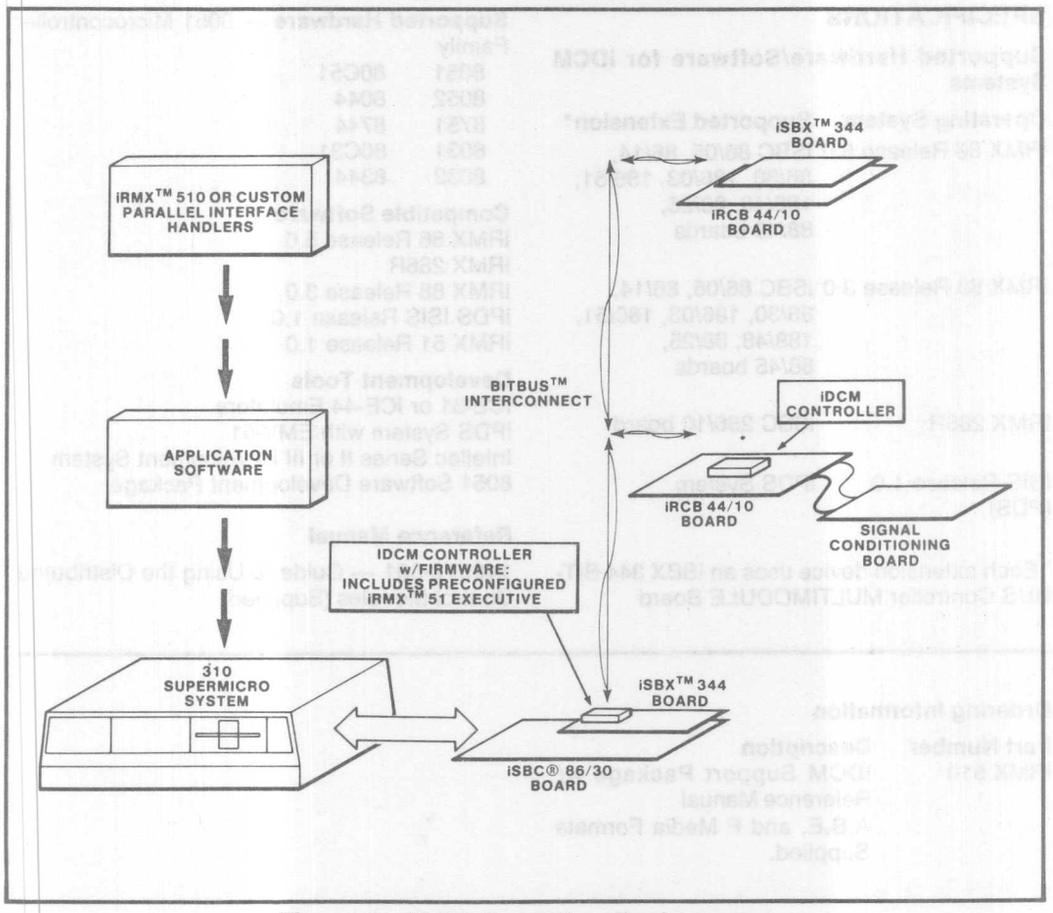


Figure 2. iDCM Operating Environment

The software handlers ease integration of other processors into an iDCM system and provide the tools to quickly expand a MULTIBUS system, or an iPDS ISIS system. Significant reduction in application system software development time results, with more effort concentrated on the overall application.

iDCM Controller Firmware

Also included in the iRMX 510 Support Package is the iDCM Controller firmware in loadable object files, iRMX 51 libraries, and iDCM Controller Include files. An Intellec Development System and ICE-44 Emulator can be used with the loadable object files to accurately simulate the iDCM Controller. This capability significantly decreases development effort by reducing trial and error production of application system software. The iRMX 51 Interface Library and iDCM Controller Include files allow development of user code for iDCM systems.

DEVELOPMENT ENVIRONMENT

The iRMX 510 Support Package completes the development environment for iDCM application system development when used with an Intellec Series II or III Development System, In-Circuit Emulator (ICE-44), and the 8051 Software Development Package. As part of Intel's complete development environment for the 8051 family of microcontrollers, the iRMX 510 Support Package may also be used with an iPDS system and EMV-51 or an Intellec Series II or III Development System and an ICE-51 Emulator.

SPECIFICATIONS

Supported Hardware/Software for iDCM Systems

Operating System	Supported Extension*
iRMX 86 Release 5.0	iSBC 86/05, 86/14, 86/30, 186/03, 186/51, 188/48, 88/25, 88/45 boards
iRMX 88 Release 3.0	iSBC 86/05, 86/14, 86/30, 186/03, 186/51, 188/48, 88/25, 88/45 boards
iRMX 286R	iSBC 286/10 board
ISIS Release 1.0 (PDS)	iPDS System

*Each extension device uses an iSBX 344 BITBUS Controller MULTIMODULE Board

Supported Hardware — 8051 Microcontroller Family

8051	80C51
8052	8044
8751	8744
8031	80C31
8032	8344

Compatible Software
iRMX 86 Release 5.0
iRMX 286R
iRMX 88 Release 3.0
iPDS ISIS Release 1.0
iRMX 51 Release 1.0

Development Tools
ICE-51 or ICE-44 Emulators
iPDS System with EMV-51
Intellec Series II or III Development System
8051 Software Development Package

Reference Manual

146312-001 — Guide to Using the Distributed Control Modules (Supplied)

Ordering Information

Part Number	Description
iRMX 510	iDCM Support Package w/ Reference Manual A,B,E, and F Media Formats Supplied.

8044AH/8344AH
HIGH PERFORMANCE 8-BIT MICROCONTROLLER WITH ON-CHIP SERIAL COMMUNICATION CENTER

- **8044AH — CPU/SIU with Factory Mask Programmable ROM**
- **8344AH — An 8044AH used with External Program Memory**
- **8744H — An 8044AH with User Programmable/Erasable EPROM**

8051 MICROCONTROLLER CORE

- **Optimized for Real Time Control 12 MHz Clock, Priority Interrupts, 32 Programmable I/O lines, Two 16-bit Timer/Counters**
- **Boolean Processor**
- **4K x 8 ROM, 192 x 8 RAM**
- **64K Accessible External Program Memory**
- **64K Accessible External Data Memory**
- **4 μs Multiply and Divide**

SERIAL INTERFACE UNIT (SIU)

- **Serial Communication Processor that Operates Concurrently to C.P.U.**
- **2.4 Mbps Maximum Data Rate**
- **375 Kbps using On-Chip Phase Locked Loop**
- **Communication Software in Silicon:**
  - **Complete Data Link Functions**
  - **Automatic Station Responses**
- **Operates as an SDLC Primary or Secondary Station**

The RUPI-44 family integrates a high performance 8-bit Microcontroller, the Intel 8051 Core, with an intelligent/high performance HDLC/SDLC serial communication controller, called the Serial Interface Unit (SIU). See Figure 1. This dual architecture allows complex control and high speed data communication functions to be realized cost effectively.

Specifically, the 8044's Microcontroller features: 4K byte On-Chip program memory space; 32 I/O lines; two 16-bit timer/event counters; a 5-source; 2-level interrupt structure; a full duplex serial channel; a Boolean processor; and on-chip oscillator and clock circuitry. Standard TTL and most byte-oriented MCS-80 and MCS-85 peripherals can be used for I/O and memory expansion.

The Serial Interface Unit (SIU) manages the interface to a high speed serial link. The SIU offloads the On-Chip 8051 Microcontroller of communication tasks, thereby freeing the CPU to concentrate on real time control tasks.

The RUPI-44 family consists of the 8044, 8744, and 8344. All three devices are identical except in respect of on-chip program memory. The 8044 contains 4K bytes of mask-programmable ROM. User programmable EPROM replaces ROM in the 8744. The 8344 addresses all program memory externally.

The RUPI-44 devices are fabricated with Intel's reliable +5 volt, silicon-gate HMOSII technology and packaged in a 40-pin DIP.

8044's Dual Controller Architecture

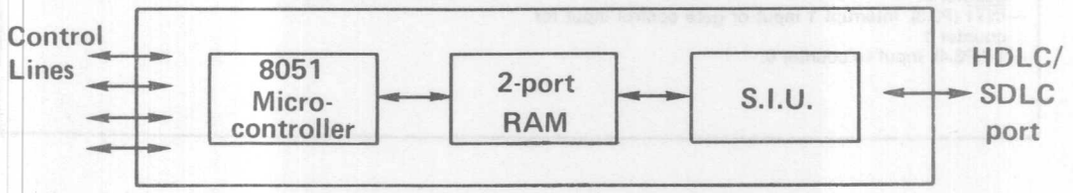


Figure 1. Dual Controller Architecture

Table 1. RUPI™ -44 Family Pin Description

VSS
Circuit ground potential.

VCC
+5V power supply during operation and program verification.

PORT 0
Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.

PORT 1
Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.

In non-loop mode two of the I/O lines serve alternate functions:
— $\overline{RTS}$ (P1.6). Request-to-Send output. A low indicates that the RUPI-44 is ready to transmit.
— $\overline{CTS}$ (P1.7) Clear-to-Send input. A low indicates that a receiving station is ready to receive.

PORT 2
Port 2 is an 8-bit quasi-bidirection I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.

PORT 3
Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads.

In addition to I/O, some of the pins also serve alternate functions as follows:
— I/O RxD (P3.0). In point-to-point or multipoint configurations, this pin controls the direction of pin P3.1. Serves as Receive Data input in loop and diagnostic modes.
— DATA TxD (P3.1) In point-to-point or multipoint configurations, this pin functions as data input/output. In loop mode, it serves as transmit pin. A '0' written to this pin enables diagnostic mode.
— $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.
— $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.
— T0(P3.4). Input to counter 0.
— SCLK T1 (P3.5). In addition to I/O, this pin provides input to counter 1 or serves as SCLK (serial clock) input.
— $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.
— $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST/VPD
A high level on this pin resets the RUPI-44. A small internal pulldown resistor permits power-on reset using only a capacitor connected to VCC. If VPD is held within its spec while VCC drops below spec, VPD will provide standby power to the RAM. When VPD is low, the RAM's current is drawn from VCC.

ALE/$\overline{PROG}$
Provides Address Latch Enable output used for latching the address into external memory during normal operation. It is activitated every six oscillator periods except during an external data memory access. It also receives the program pulse input for programming the EPROM version.

$\overline{PSEN}$
The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

$\overline{EA}$/VPP
When held at a TTL high level, the RUPI-44 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the RUPI-44 fetches all instructions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage on the 8744.

XTAL 1
Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to VSS when external source is used on XTAL 2.

XTAL 2
Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

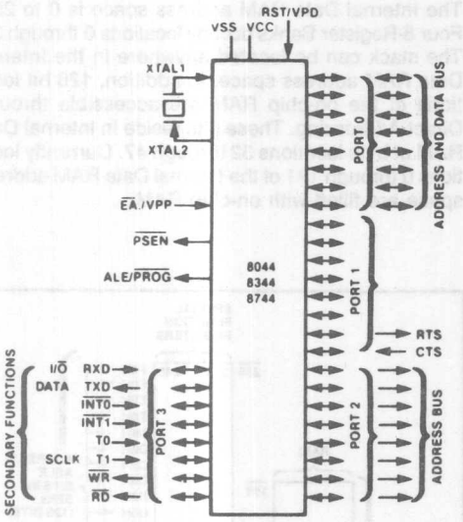


Figure 2.
Logic Symbol

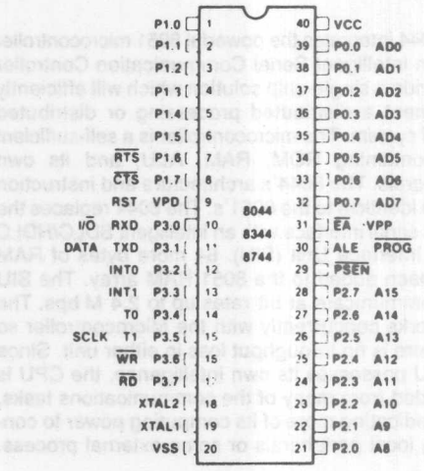


Figure 3. Pin Configuration

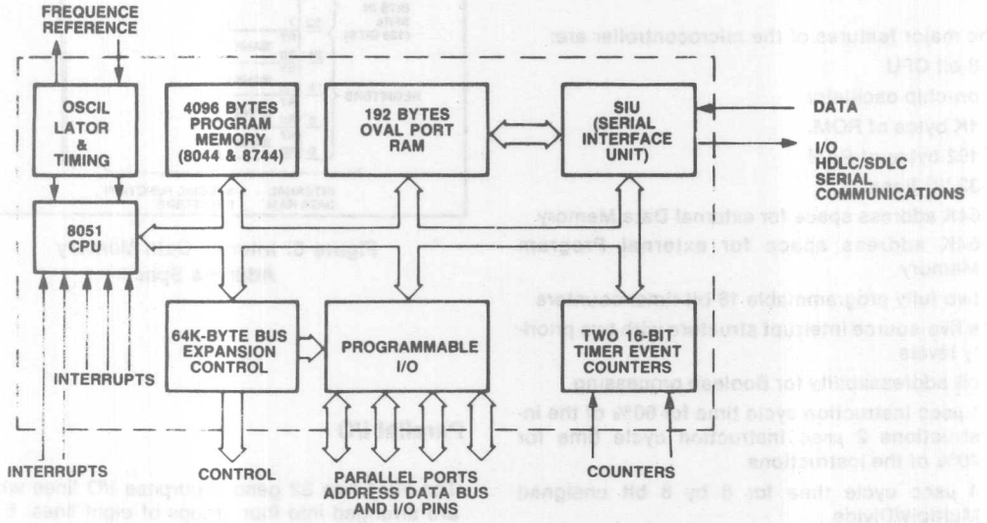


Figure 4.
Block Diagram

Functional Description

General

The 8044 integrates the powerful 8051 microcontroller with an intelligent Serial Communication Controller to provide a single-chip solution which will efficiently implement a distributed processing or distributed control system. The microcontroller is a self-sufficient unit containing ROM, RAM, ALU, and its own peripherals. The 8044's architecture and instruction set are identical to the 8051's. The 8044 replaces the 8051's serial interface with an intelligent SDLC/HDLC Serial Interface Unit (SIU). 64 more bytes of RAM have been added to the 8051 RAM array. The SIU can communicate at bit rates up to 2.4 M bps. The SIU works concurrently with the Microcontroller so that there is no throughput loss in either unit. Since the SIU possesses its own intelligence, the CPU is off-loaded from many of the communications tasks, thus dedicating more of its computing power to controlling local peripherals or some external process.

The Microcontroller

The microcontroller is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time application such as instrumentation, industrial control, and intelligent computer peripherals.

The major features of the microcontroller are:

- 8-bit CPU
- on-chip oscillator
- 4K bytes of ROM
- 192 bytes of RAM
- 32 I/O lines
- 64K address space for external Data Memory
- 64K address space for external Program Memory
- two fully programmable 16-bit timer/counters
- a five-source interrupt structure with two priority levels
- bit addressability for Boolean processing
- 1 μsec instruction cycle time for 60% of the instructions 2 μsec instruction cycle time for 40% of the instructions
- 4 μsec cycle time for 8 by 8 bit unsigned Multiply/Divide

Internal Data Memory

Functionally the Internal Data Memory is the most flexible of the address spaces. The Internal Data Memory space is subdivided into a 256-byte Internal Data RAM address space and a 128-byte Special Function Register address space as shown in Figure 5.

The Internal Data RAM address space is 0 to 255. Four 8-Register Banks occupy locations 0 through 31. The stack can be located anywhere in the Internal Data RAM address space. In addition, 128 bit locations of the on-chip RAM are accessible through Direct Addressing. These bits reside in Internal Data RAM at byte locations 32 through 47. Currently locations 0 through 191 of the Internal Data RAM address space are filled with on-chip RAM.

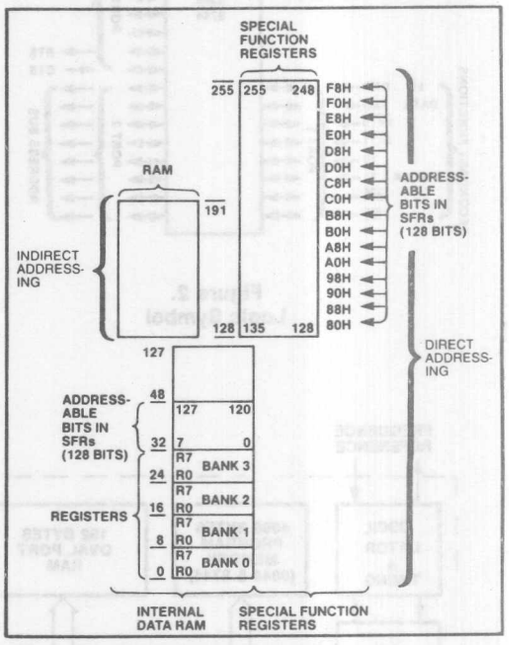


Figure 5. Internal Data Memory Address Space

Parallel I/O

The 8044 has 32 general-purpose I/O lines which are arranged into four groups of eight lines. Each group is called a port. Hence there are four ports; Port 0, Port 1, Port 2, and Port 3. Up to five lines from 1 and Port 2 are dedicated to supporting the serial channel when the SIU is invoked. Due to the nature of the serial port, two of Port 3's I/O lines (P3.0 and P3.1) do not have latched outputs. This is true whether or not the serial channel is used.

Table 1. MCS®-51 Instruction Set Description

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

LOGICAL OPERATIONS

Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1

LOGICAL OPERATIONS (CONTINUED)

Mnemonic		Destination	Byte	Cyc
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Mov immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2

Table 1. (Cont.)

DATA TRANSFER (CONTINUED)

Mnemonic		Description	Byte	Cyc
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order Digit ind RAM w A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Byte	Cyc
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp, immed, to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp, immed, to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Comp, immed, to ind, & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn —Working register R0-R7
direct —128 internal RAM locations, any I/O port, control or status register
@Ri —Indirect internal RAM location addressed by register R0 or R1
#data —8-bit constant included in instruction
#data16 —16-bit constant included as bytes 2 & 3 of instruction
bit —128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16 —Destination address for LCALL & LJMP may be anywhere within the 64-K program memory address space
Addr11 —Destination address for ACALL & AJMP will be within the same 2-K page of program memory as the first byte of the following instruction
rel —SJMP and all conditional jumps include an 8-bit offset byte, Range is +127-128 bytes relative to first byte of the following instruction

All mnemonics copyrighted © Intel Corporation 1979

Port 0 and Port 2 also have an alternate dedicated function. When placed in the external access mode, Port 0 and Port 2 become the means by which the 8044 communicates with external program memory. Port 0 and Port 2 are also the means by which the 8044 communicates with external data memory. Peripherals can be memory mapped into the address space and controlled by the 8044.

Timer/Counters

The 8044 contains two 16-bit counters which can be used for measuring time intervals, measuring pulse widths, counting events, generating precise periodic interrupt requests, and clocking the serial communications. Internally the Timers are clocked at 1/12 of the crystal frequency, which is the instruction cycle time. Externally the counters can run up to 500 KHz.

Interrupt System

External events and the real-time driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two priority level, nested interrupt system is provided. Interrupt response latency ranges from 3 μsec to 7 μsec when using a 12 MHz clock.

All five interrupt sources can be mapped into one of the two priority levels. Each interrupt source can be enabled or disabled individually or the entire interrupt system can be enabled or disabled. The five interrupt sources are: Serial Interface Unit, Timer 1, Timer 2, and two external interrupts. The external interrupts can be either level or edge triggered.

Serial Interface Unit (SIU)

The Serial Interface Unit is used for HDLC/SDLC communications. It handles Zero Bit Insertion/Deletion, Flags, automatic address recognition, and a 16-bit cyclic redundancy check. In addition it implements in hardware a subset of the SDLC protocol such that it responds to many SDLC frames without CPU intervention. In certain applications it is advantageous to have the CPU control the reception or transmission of every single frame. For this reason the SIU has two modes of operation: "AUTO" and "FLEXIBLE" (or "NON-AUTO"). It is in the AUTO mode that the SIU responds to SDLC frames without CPU intervention; whereas, in the FLEXIBLE mode the reception or transmission of every single frame will be under CPU control.

There are three control registers and eight parameter registers that are used to operate the serial interface. These registers are shown in Figure 5 and Figure 6. The control registers set the modes of operation and provide status information. The eight parameter registers buffer the station address, receive and transmit control bytes, and point to the on-chip transmit and receive buffers.

Data to be received or transmitted by the SIU must be buffered anywhere within the 192 bytes of on-chip RAM. Transmit and receive buffers are not allowed to "wrap around" in RAM; a "buffer end" is generated after address 191 is reached.

AUTO Mode

In the AUTO mode the SIU implements in hardware a subset of the SDLC protocol such that it responds to many SDLC frames without CPU intervention. All AUTO mode responses to the primary station will conform to IBM's SDLC definition. The advantages of the AUTO mode are that less software is required to implement a secondary station, and the hardware generated response to polls is much faster than doing it in software. However, the Auto mode can not be used at a primary station.

To transmit in the AUTO mode the CPU must load the Transmit Information Buffer, Transmit Buffer Start register, Transmit Buffer Length register, and set the Transmit Buffer Full bit. The SIU automatically responds to a poll by transmitting an information frame with the P/F bit in the control field set. When the SIU receives a positive acknowledgement from the primary station, it automatically increments the Ns field in the NSNR register and interrupts the CPU. A negative acknowledgement would cause the SIU to retransmit the frame.

To receive in the AUTO mode, the CPU loads the Receive Buffer Start register, the Receive Buffer Length register, clears the Receive Buffer Protect bit, and sets the Receive Buffer Empty bit. If the SIU is polled in this state, and the TBF bit indicates that the Transmit Buffer is empty, an automatic RR response will be generated. When a valid information frame is received the SIU will automatically increment Nr in the NSNR register and interrupt the CPU.

While in the AUTO mode the SIU can recognize and respond to the following commands without CPU intervention: I (Information), RR (Receive Ready), RNR (Receive Not Ready), REJ (Reject), and UP (Unnumbered Poll). The SIU can generate

REGISTER NAMES	SYMBOLIC ADDRESS	BIT ADDRESS	BYTE ADDRESS
B REGISTER	B	247 through 240	240 (F0H)
ACCUMULATOR	ACC	231 through 224	224 (E0H)
*THREE BYTE FIFO	FIFO		223 (DFH)
	FIFO		222 (DEH)
	FIFO		221 (DDH)
TRANSMIT BUFFER START	TBS		220 (DCH)
TRANSMIT BUFFER LENGTH	TBL		219 (DBH)
TRANSMIT CONTROL BYTE	TCB		218 (DAH)
* SIU STATE COUNTER	SIUST		217 (D9H)
SEND COUNT RECEIVE COUNT	NSNR	223 through 216	216 (D8H)
PROGRAM STATUS WORD	PSW	215 through 208	208 (D0H)
*DMA COUNT	DMA CNT		207 (CFH)
STATION ADDRESS	STAD		206 (CEH)
RECEIVE FIELD LENGTH	RFL		205 (CDH)
RECEIVE BUFFER START	RBS		204 (CCH)
RECEIVE BUFFER LENGTH	RBL		203 (CBH)
RECEIVE CONTROL BYTE	RCB		202 (CAH)
SERIAL MODE	SMD		201 (C9H)
STATUS REGISTER	STS	207 through 200	200 (C8H)
INTERRUPT PRIORITY CONTROL	IP	191 through 184	184 (B8H)
PORT 3	P3	183 through 176	176 (B0H)
INTERRUPT ENABLE CONTROL	IE	175 through 168	168 (A8H)
PORT 2	P2	167 through 160	160 (A0H)
PORT 1	P1	151 through 144	144 (90H)
TIMER HIGH 1	TH1		141 (8DH)
TIMER HIGH 0	TH0		140 (8CH)
TIMER LOW 1	TL1		139 (8BH)
TIMER LOW 0	TL0		138 (8AH)
TIMER MODE	TMOD		137 (89H)
TIMER CONTROL	TCON	143 through 136	136 (88H)
DATA POINTER HIGH	DPH		131 (83H)
DATA POINTER LOW	DPL		130 (82H)
STACK POINTER	SP		129 (81H)
PORT 0	P0	135 through 128	128 (80H)

SFR's CONTAINING DIRECT ADDRESSABLE BITS

*ICE Support Hardware registers. Under normal operating conditions there is no need for the CPU to access these registers.

Figure 5. Mapping of Special Function registers

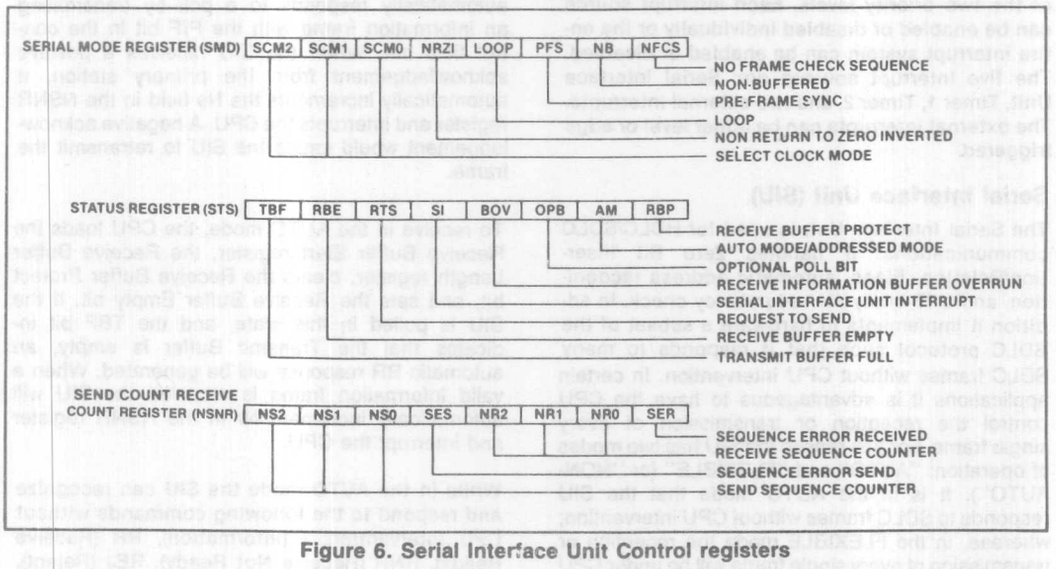


Figure 6. Serial Interface Unit Control registers

the following responses without CPU intervention: I (Information), RR (Receive Ready), and RNR (Receive Not Ready).

When the Receive Buffer Empty bit (RBE) indicates that the Receive Buffer is empty, the receiver is enabled, and when the RBE bit indicates that the Receive Buffer is full, the receiver is disabled. Assuming that the Receive Buffer is empty, the SIU will respond to a poll with an I frame if the Transmit Buffer is full. If the Transmit Buffer is empty, the SIU will respond to a poll with a RR command if the Receive Buffer Protect bit (RBP) is cleared, or an RNR command if RBP is set.

FLEXIBLE (or NON-AUTO) Mode

In the FLEXIBLE mode all communications are under control of the CPU. It is the CPU's task to encode and decode control fields, manage acknowledgements, and adhere to the requirements of the HDLC/SDLC protocols. The 8044 can be used as a primary or a secondary station in this mode.

To receive a frame in the FLEXIBLE mode, the CPU must load the Receive Buffer Start register, the Receive Buffer Length register, clear the Receive Buffer Protect bit, and set the Receive Buffer Empty bit. If a valid opening flag is received and the address field matches the byte in the Station Address register or the address field contains a broadcast address, the 8044 loads the control field in the receive control byte register, and loads the I field in the receive buffer. If there is no CRC error, the SIU interrupts the CPU, indicating a frame has just been received. If there is a CRC error, no interrupt occurs. The Receive Field Length register provides the number of bytes that were received in the information field.

To transmit a frame, the CPU must load the transmit information buffer, the Transmit Buffer Start register, the Transmit Buffer Length register, the Transmit Control Byte, and set the TBF and the RFS bit. The SIU, unsolicited by an HDLC/SDLC frame, will transmit the entire information frame, and interrupt the CPU, indicating the completion of transmission. For supervisory frames or unnumbered frames, the transmit buffer length would be 0.

CRC

The FCS register is initially set to all 1's prior to calculating the FCS field. The SIU will not interrupt the CPU if a CRC error occurs (in both AUTO and FLEXIBLE modes). The CRC error is cleared upon receiving of an opening flag.

Frame Format Options

In addition to the standard SDLC frame format, the 8044 will support the frames displayed in Figure 7. The standard SDLC frame is shown at the top of this figure. For the remaining frames the information field will incorporate the control or address bytes and the frame check sequences; therefore these fields will be stored in the Transmit and Receive buffers. For example, in the non-buffered mode the third byte is treated as the beginning of the information field. In the non-addressed mode, the information field begins after the opening flag. The mode bits to set the frame format options are found in the Serial Mode register and the Status register.

EXTENDED ADDRESSING

To realize an extended control field or an extended address field using the HDLC protocol, the FLEXIBLE mode must be used. For an extended control field, the SIU is programmed to be in the non-buffered mode. The extended control field will be the first and second bytes in the Receive and Transmit Buffers. For extended addressing the SIU is placed in the non-addressed mode. In this mode the CPU must implement the address recognition for received frames. The addressing field will be the initial bytes in the Transmit and Receive buffers followed by the control field.

The SIU can transmit and receive only frames which are multiples of 8 bits. For frames received with other than 8-bit multiples, a CRC error will cause the SIU to reject the frame.

SDLC Loop Networks

The SIU can be used in a ADLC loop as a secondary or primary station. When the SIU is placed in the Loop mode it receives the data on pin 10 and transmits the data one bit time delayed on pin 11. It can also recognize the Go ahead signal and change it into a flag when it is ready to transmit. As a secondary station the SIU can be used in the AUTO or FLEXIBLE modes. As a primary station the FLEXIBLE mode is used; however, additional hardware is required for generating the Go Ahead bit pattern. In the Loop mode the maximum data rate is 1 Mbps clocked or 375 Kbps self-clocked.

SDLC Multidrop Networks

The SIU can be used in a SDLC non-loop configuration as a secondary or primary station. When the SIU is placed in the non-loop mode, data is received and transmitted on pin 11, and pin 10 drives a tri-state buffer. In non-loop mode, modem interface pins, RTS and CTS, become available.

FRAME OPTION	NFCS	NB	AM'	FRAME FORMAT
Standard SDLC NON-AUTO Mode	0	0	0	F A C I FCS F
Standard SDLC AUTO Mode	0	0	1	F A C I FCS F
Non-Buffered Mode NON-AUTO Mode	0	1	1	F A I FCS F
Non-Addressed Mode NON-AUTO Mode	0	1	0	F I FCS F
No FCS Field NON-AUTO Mode	1	0	0	F A C I F
No FCS Field AUTO Mode	1	0	1	F A C I F
No FCS Field Non-Buffered Mode NON-AUTO Mode	1	1	1	F A I F
No FCS Field Non-Addressed Mode NON-AUTO Mode	1	1	0	F I F

Mode Bits:
AM — "AUTO" Mode/Addressed Mode
NB — Non-Buffered Mode
NFCS — No FCS Field Mode

Key to Abbreviations:
F = Flag (01111110)
A = Address Field
C = Control Field
I = Information Field
FCS = Frame Check Sequence

Note 1: The AM bit function is controlled by the NB bit. When NB = 0, AM becomes AUTO mode select, when NB = 1, AM becomes Address mode select.

Figure 7. Frame Format Options

Data Clocking Options

The 8044's serial port can operate in an externally clocked or self clocked system. A clocked system provides to the 8044 a clock synchronization to the data. A self-clocked system uses the 8044's on-chip Digital Phase Locked Loop (DPLL) to recover the clock from the data, and clock this data into the Serial Receive Shift Register.

In this mode, a clock synchronized with the data is externally fed into the 8044. This clock may be generated from an External Phase Locked Loop, or possibly supplied along with the data. The 8044 can transmit and receive data in this mode at rates up to 2.4 Mbps.

This self clocked mode allows data transfer without a common system data clock. An on-chip Digital Phase Locked Loop is employed to recover the data clock which is encoded in the data stream. The DPLL will converge to the nominal bit center within eight bit transitions, worst case. The DPLL requires a reference clock of either 16 times (16x) or 32 times (32x) the data rate. This reference clock may be externally applied or internally generated. When internally generated either the 8044's internal logic clock (crystal frequency divided by two) or the timer 1 overflow is used as the reference clock. Using the internal timer 1 clock the data reates can vary from 244 to 62.5 Kbps. Using the internal logic clock at a 16x sampling rate, receive data can either be 187.5 Kbps, or 375 Kbps. When the reference clock for the DPLL is externally applied the data rates can vary from 0 to 375 Kbps at a 16x sampling rate.

To aid in a Phase Locked Loop capture, the SIU has a NRZI (Non Return to Zero Inverted) data encoding and decoding option. Additionally the SIU has a preframe sync option that transmits two bytes of alternating 1's and 0's to ensure that the receive station DPLL will be synchronized with the data by the time it receives the opening flag.

Control and Status Registers

There are three SIU Control and Status Registers:

Serial Mode Register (SMD)

Status/Command Register (STS)

Send/Receive Count Register (NSNR)

The SMD, STS, and NSNR registers are all cleared by system reset. This assures that the SIU will power up in an idle state (neither receiving nor transmitting).

These registers and their bit assignments are described below.

SMD: Serial Mode Register (byte-addressable)

Bit: 7 6 5 4 3 2 1 0

SCM2	SCM1	SCM0	NRZI	LOOP	PFS	NB	NFCS

The Serial Mode Register (Address C9H) selects the operational modes of the SIU. The 8044 CPU can both read and write SMD. The SIU can read SMD but cannot write to it. To prevent conflict between CPU and SIU access to SMD, the CPU should write SMD only when the Request To Send (RTS) and Receive Buffer Empty (RBE) bits (in the STS register) are both false (0). Normally, SMD is accessed only during initialization.

The individual bits of the Serial Mode Register are as follows:

Bit #	Name	Description
SMD.0	NFCS	No FCS field in the SDLC frame.
SMD.1	NB	Non-Buffered mode. No control field in the SDLC frame.
SMD.2	PFS	Pre-Frame Sync mode. In this mode, the 8044 transmits two bytes before the first flag of a frame, for DPLL synchronization. If NRZI is enabled, 00H is sent; otherwise, 55H is sent. In either case, 16 pre-frame transitions are guaranteed.
SMD.3	LOOP	Loop configuration.
SMD.4	NRZI	NRZI coding option.
SMD.5	SCM0	Select Clock Mode — Bit 0
SMD.6	SCM1	Select Clock Mode — Bit 1
SMD.7	SCM2	Select Clock Mode — Bit 2

The SCM bits decode as follows:

SCM 2	SCM 1	SCM 0	Clock Mode	Data Rate (Bits/sec)*
0	0	0	Externally clocked	0-2.4M**
0	0	1	Reserved	
0	1	0	Self clocked, timer overflow	244-62.5K
0	1	1	Reserved	
1	0	0	Self clocked, external 16x	0-375K
1	0	1	Self clocked, external 32x	0-187.5K
1	1	0	Self clocked, internal fixed	375K
1	1	1	Self clocked, internal fixed	187.5k

*Based on a 12 Mhz crystal frequency
**0-1M bps in loop configuration

STS: Status/Command Register (bit-addressable)

Bit: 7 6 5 4 3 2 1 0

TBF	RBE	RTS	SI	BOV	OPB	AM	RBP

The Status/Command Register (Address C8H) provides operational control of the SIU by the 8044 CPU, and enables the SIU to post status information for the CPU's access. The SIU can read STS, and can alter certain bits, as indicated below. The CPU can both read and write STS asynchronously. However, 2-cycle instructions that access STS during both cycles ('JBC/B, REL' and 'MOV /B,C.') should not be used, since the SIU may write to STS between the two CPU accesses.

The individual bits of the Status/Command Register are as follows:

Bit #	Name	Description
STS.0	RBP	Receive Buffer Protect. Inhibits writing of data into the receive buffer. In AUTO mode, RBP forces an RNR response instead of an RR.
STS.1	AM	AUTO Mode/Addressed Mode. Selects AUTO mode where AUTO mode is allowed. If NB is true, (=1), the AM bit selects the addressed mode. AM may be cleared by the SIU.
STS.2	OPB	Optional Poll Bit. Determines whether the SIU will generate an AUTO response to an optional poll (UP with P=0). OPB may be set or cleared by the SIU.
STS.3	BOV	Receive Buffer Overrun. BOV may be set or cleared by the SIU.
STS.4	SI	SIU Interrupt. This is one of the five interrupt sources to the CPU. The vector location = 23H. SI may be set by the SIU. It should be cleared by the CPU before returning from an interrupt routine.
STS.5	RTS	Request To Send. Indicates that the 8044 is ready to transmit or is transmitting. RTS may be read or written by the CPU. RTS may be read by the SIU, and in AUTO mode may be written by the SIU.

STS.6	RBE	Receive Buffer Empty. RBE can be thought of as Receive Enable. RBE is set to one by the CPU when it is ready to receive a frame, or has just read the buffer, and to zero by the SIU when a frame has been received.
STS.7	TBF	Transmit Buffer Full. Written by the CPU to indicate that it has filled the transmit buffer. TBF may be cleared by the SIU.

NSNR: Send/Receive Count Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	NS2	NS1	NS0	SES	NR2	NR1	NR0	SER

The Send/Receive Count Register (Address D8H) contains the transmit and receive sequence numbers, plus tally error indications. The SIU can both read and write NSNR. The 8044 CPU can both read and write NSNR asynchronously. However, 2-cycle instructions that access NSNR during both cycles ('JBC /B, REL', and 'MOV /B,C') should not be used, since the SIU may write to NSNR between the two 8044 CPU accesses.

The individual bits of the Send/Receive Count Register are as follows:

Bit #	Name	Description
NSNR.0	SER	Receive Sequence Error: NS (P) ≠ NR (S)
NSNR.1	NR0	Receive Sequence Counter—Bit 0
NSNR.2	NR1	Receive Sequence Counter—Bit 1
NSNR.3	NR2	Receive Sequence Counter—Bit 2
NSNR.4	SES	Send Sequence Error: NP (P) ≠ NS (S) and NP (P) ≠ NS (S) + 1
NSNR.5	NS0	Send Sequence Counter — Bit 0
NSNR.6	NS1	Send Sequence Counter — Bit 1
NSNR.7	NS2	Send Sequence Counter — Bit 2

Parameter Registers

There are eight parameter registers that are used in connection with SIU operation. All eight registers may be read or written by the 8044 CPU. RFL and RCB are normally loaded by the SIU.

The eight parameter registers are as follows:

STAD: Station Address Register (byte-addressable)

The Station Address register (Address CEH) contains the station address. To prevent access conflict, the CPU should access STAD only when the SIU is idle (RTS=0 and RBE=0). Normally, STAD is accessed only during initialization.

TBS: Transmit Buffer Start Address Register (byte-addressable)

The Transmit Buffer Start address register (Address DCH) points to the location in on-chip RAM for the beginning of the I-field of the frame to be transmitted. The CPU should access TBS only when the SIU is not transmitting a frame (when TBF=0).

TBL: Transmit Buffer Length Register (byte-addressable)

The Transmit Buffer Length register (Address DBH) contains the length (in bytes) of the I-field to be transmitted. A blank I-field (TBL=0) is valid. The CPU should access TBL only when the SIU is not transmitting a frame (when TBF=0).

NOTE: The transmit and recieve buffers are not allowed to "wrap around" in the on-chip RAM. A "buffer end" is automatically generated if address 191 (BFH) is reached.

TCB: Transmit Control Byte Register (byte-addressable)

The Transmit Control Byte register (Address DAH) contains the byte which is to be placed in the control field of the transmitted frame, during NON-AUTO mode transmission. The CPU should access TCB only when the SIU is not transmitting a frame (when TBF=0). The N_S and N_R counters are not used in the NON-AUTO mode.

RBS: Receive Buffer Start Address Register (byte-addressable)

The Receive Buffer Start address register (Address CCH) points to the location in on-chip RAM where the beginning of the I-field of the frame being received is to be stored. The CPU should write RBS only when the SIU is not receiving a frame (when RBE=0).

RBL: Receive Buffer Length Register (byte-addressable)

The Receive Buffer Length register (Address CBH) contains the length (in bytes) of the area in on-chip RAM allocated for the received I-field. RBL=0 is valid. The CPU should write RBL only when RBE=0.

RFL: Receive Field Length Register (byte-addressable)

The Received Field Length register (Address CDH) contains the length (in bytes) of the received I-field that has just been loaded into on-chip RAM. RFL is loaded by the SIU. RFL=0 is valid. RFL should be accesssed by the CPU only when RBE=0.

RCB: Receive Control Byte Register (byte-addressable)

The Received Control Byte register (Address CAH) contains the control field of the frame that has just been received. RCB is loaded by the SIU. The CPU can only read RCB, and should only access RCB when RBE=0.

ICE Support Registers

The 8044 In-Circuit Emulator (ICE-44) allows the user to exercise the 8044 application system and monitor the execution of instructions in real time.

The emulator operates with Intel's Intellec® development system. The development system interfaces with the user's 8044 system through an in-cable buffer box. The cable terminates in a 8044 pin-compatible plug, which fits into the 8044 socket in the user's system. With the emulator plug in place, the user can excercise his system in real time while collecting up to 255 instruction cycles of real-time data. In addition, he can single-step the program.

Static RAM is available (in the in-cable buffer box) to emulate the 8044 internal and external program memory and external data memory. The designer can display and alter the contents of the replacement memory in the buffer box, the internal data memory, and the internal 8044 registers, including the SFRs.

SIUST: SIU State Counter (byte-addressable)

The SIU State Counter (Address D9H) reflects the state of the internal logic which is under SIU control. Therefore, care must be taken not to write into this register. This register provides a useful means for debugging 8044 receiver problem.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0 to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin With
Respect to Ground (V_{SS}) . . . . . −0.5V to +7V
Power Dissipation . . . . . 2 Watts

*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, V_{SS} = 0V)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
VIL	Input Low Voltage	−0.5		0.8	V	
VIH	Input High Voltage (Except RST/VPD and XTAL2)	2.0		VCC + 0.5	V	
VIH1	Input High Voltage To RST/VPD For Reset, XTAL2	2.5			V	XTAL1 to VSS
VPD	Power Down Voltage To RST/VPD	4.5		5.5	V	VCC = 0V
VOL	Output Low Voltage Ports 1, 2, 3 (Note 1)			0.45	V	IOL = 1.6mA
VOL1	Output Low Voltage Port 0 ALE, \PSEN (Note 1)			0.45	V	IOL = 3.2mA
VOH	Outpt High Voltage Ports 1, 2, 3	2.4			V	IOH = −80μA
VOH1	Output High Voltage Port 0, ALE, \PSEN	2.4			V	IOH = −400μA
IIL	Logical 0 Input Current Ports 1, 2, 3			−800	μA	XTAL1 at VSS VIL = 0.45V
IIH1	Input High Current.T0 RST/VPD For Reset			500	μA	Vin = VCC − 1.5V
ILI	Input Leakage Current To Port 0, \EA			10	μA	0.45V < Vin < VCC
ICC	Power Supply Current		125	200	mA	TA = 25°C
IPD	Power Down Current		15	30	mA	
CIO	Capacitance of I/O Buffer			10	pF	fc = 1MHz
IIL2	Logical 0 Input Current XTAL 2			−2.5	mA	XTAL1 = VSS VIL = 0.45V

Note 1: VOL is degraded when the RUPI-44 rapidly discharges external capacitance. This A.C. noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the RUPI-44 as possible.

Datum	Emitting Ports	Degraded I/O Lines	VOL (peak) (max)
Address	P2, P0	P1, P3	.8V
Write Data	P0	P1, P3, ALE	.8V

A.C. CHARACTERISTICS (T_A 0°C to 70°C, V_{CC} = 5V ± 10%, V_{SS} = 0V, C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100pF; C_L for All Other Outputs = 80 pF)

Program Memory

Symbol	Parameter	12 MHz Clock			Variable Clock 1/TCLCL = 1.2 MHz to 12 MHz		
		Min	Max	Units	Min	Max	Units
TLHLL	ALE Pulse Width	127		ns	2TCLCL-40		ns
TAVLL	Address Setup to ALE	43		ns	TCLCL-30		ns
TLLAX[1]	Address Hold After ALE	48		ns	TCLCL-35		ns
TLLIV	ALE To Valid Instr In		233	ns		4TCLCL-100	ns
TLLPL	ALE To $\overline{PSEN}$	58		ns	TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		ns	3TCLCL-35		ns
TPLIV	$\overline{PSEN}$ To Valid Instr In		125	ns		3TCLCL-125	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns	0		ns
TPXIZ[2]	Input Instr Float After $\overline{PSEN}$		63	ns		TCLCL-20	ns
TPXAV[2]	Address Valid After $\overline{PSEN}$	75		ns	TCLCL-8		ns
TAVIV	Address To Valid Instr In		302	ns		5TCLCL-115	ns
TAZPL	Address Float To $\overline{PSEN}$	0		ns	0		ns

Notes:
1. TLLAX for access to program memory is different from TLLAX for data memory.
2. Interfacing RUPI-44 devices with float times up to 75ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

External Data Memory

Symbol	Parameter	12 MHz Clock			Variable Clock 1/TCLCL = 1.2 MHz to 12 MHz		
		Min	Max	Units	Min	Max	Units
TRLRH	$\overline{RD}$ Pulse Width	400		ns	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		ns	6TCLCL-100		ns
TLLAX[1]	Address Hold After ALE	48		ns	TCLCL-35		
TRLDV	$\overline{RD}$ To Valid Data In		250	ns		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns	0		ns
TRHDZ	Data Float After $\overline{RD}$		97	ns		2TCLCL-70	ns
TLLDV	ALE To Valid Data In		517	ns		8TCLCL-150	ns
TAVDV	Address To Valid Data In		585	ns		9TCLCL-165	ns
TLLWL	ALE To $\overline{WR}$ or $\overline{RD}$	200	300	ns	3TCLCL-50	3TCLCL + 50	ns
TAVWL	Address To $\overline{WR}$ or $\overline{RD}$	203		ns	4TCLCL-130		ns
TWHLH	$\overline{WR}$ or $\overline{RD}$ High To ALE High	43	123	ns	TCLCL-40	TCLCL + 40	ns
TDVWX	Data Valid To $\overline{WR}$ Transition	33		ns	TCLCL-50		ns
TQVWH	Data Setup Before $\overline{WR}$	433		ns	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	33		ns	TCLCL-50		ns
TRLAZ	Address Float After $\overline{RD}$		0	ns		0	ns

Note 1. TLLAX for access to program memory is different from TLLAX for access data memory.

Serial Interface

Symbol	Parameter	Min	Max	Units	
TDCY	Data Clock	420		ns	
TDCL	Data Clock Low	180		ns	
TDCH	Data Clock High	120		ns	

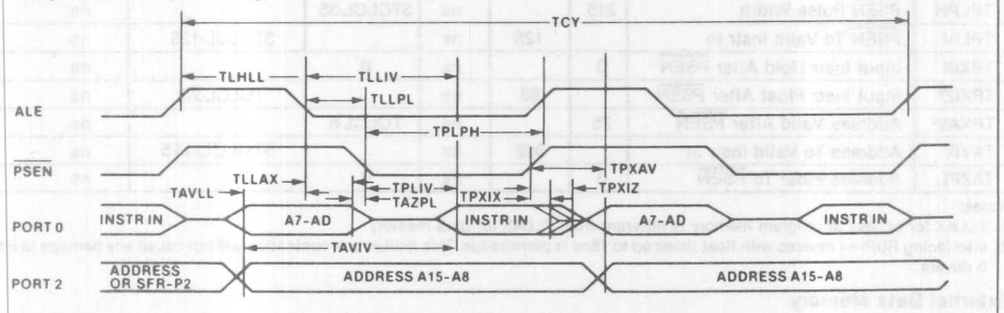
Serial Interface (Continued)

tTD	Transmit Data Delay		80	ns	
tDSS	Data Setup Time	60		ns	
tDHS	Data Hold Time	40		ns	

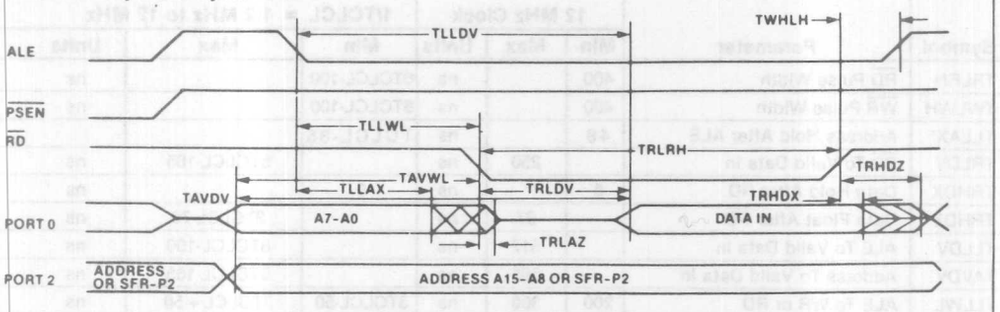
WAVEFORMS

Memory Access

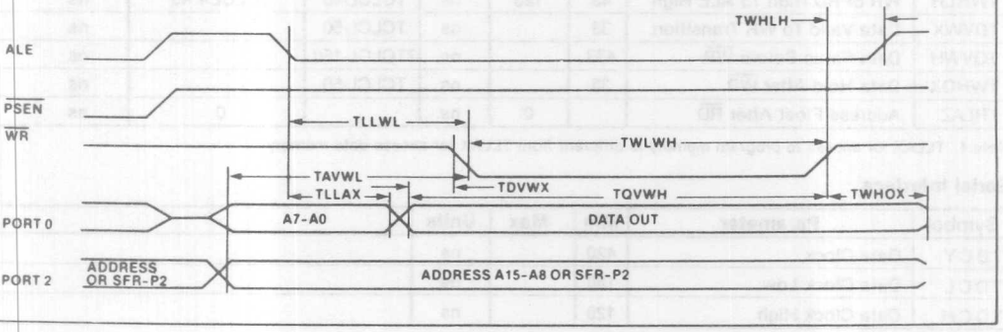
Program Memory Read Cycle



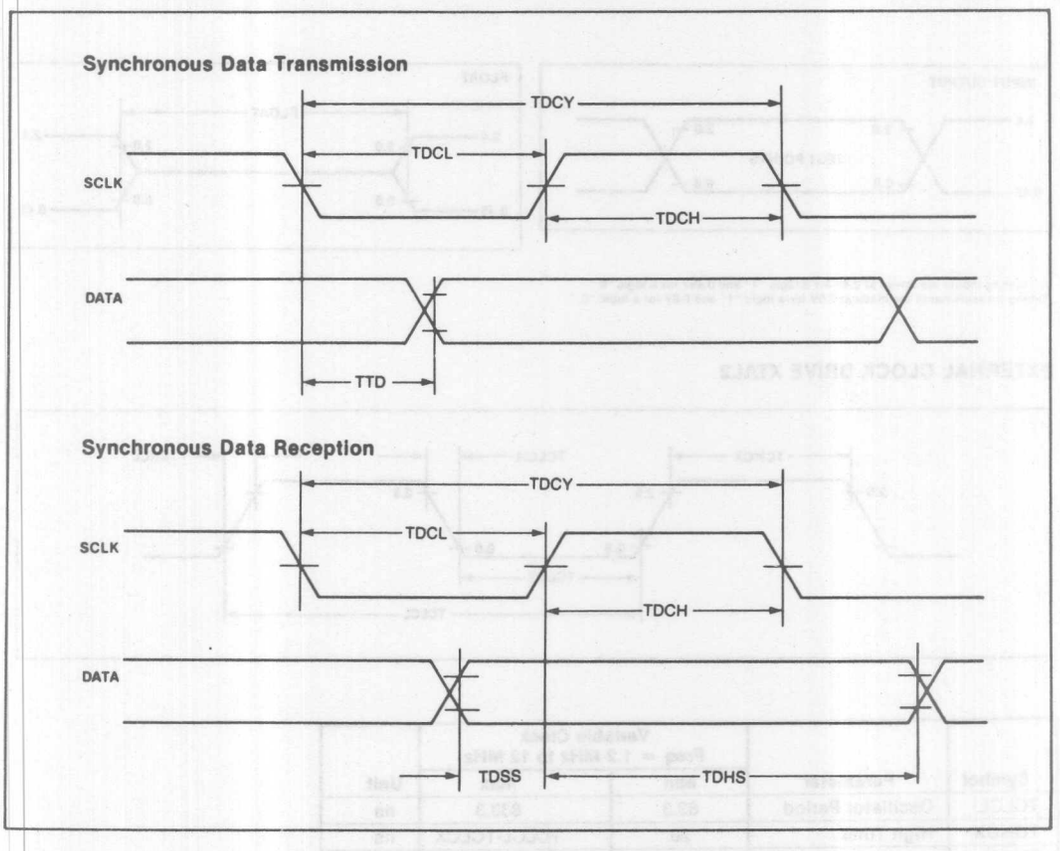
Data Memory Read Cycle



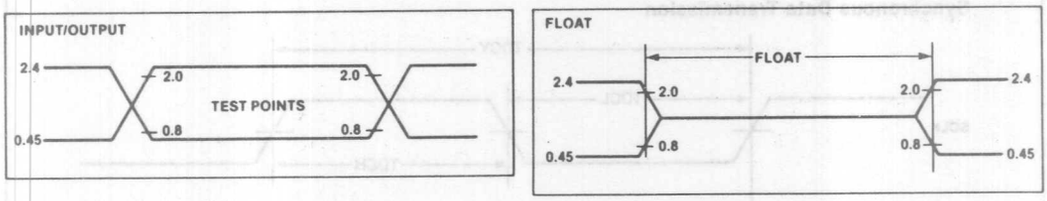
Data Memory Write Cycle



SERIAL I/O WAVEFORMS

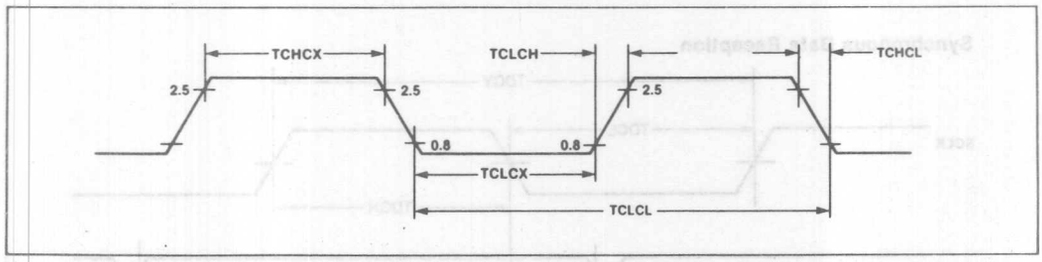


AC TESTING INPUT, OUTPUT, FLOAT WAVEFORMS



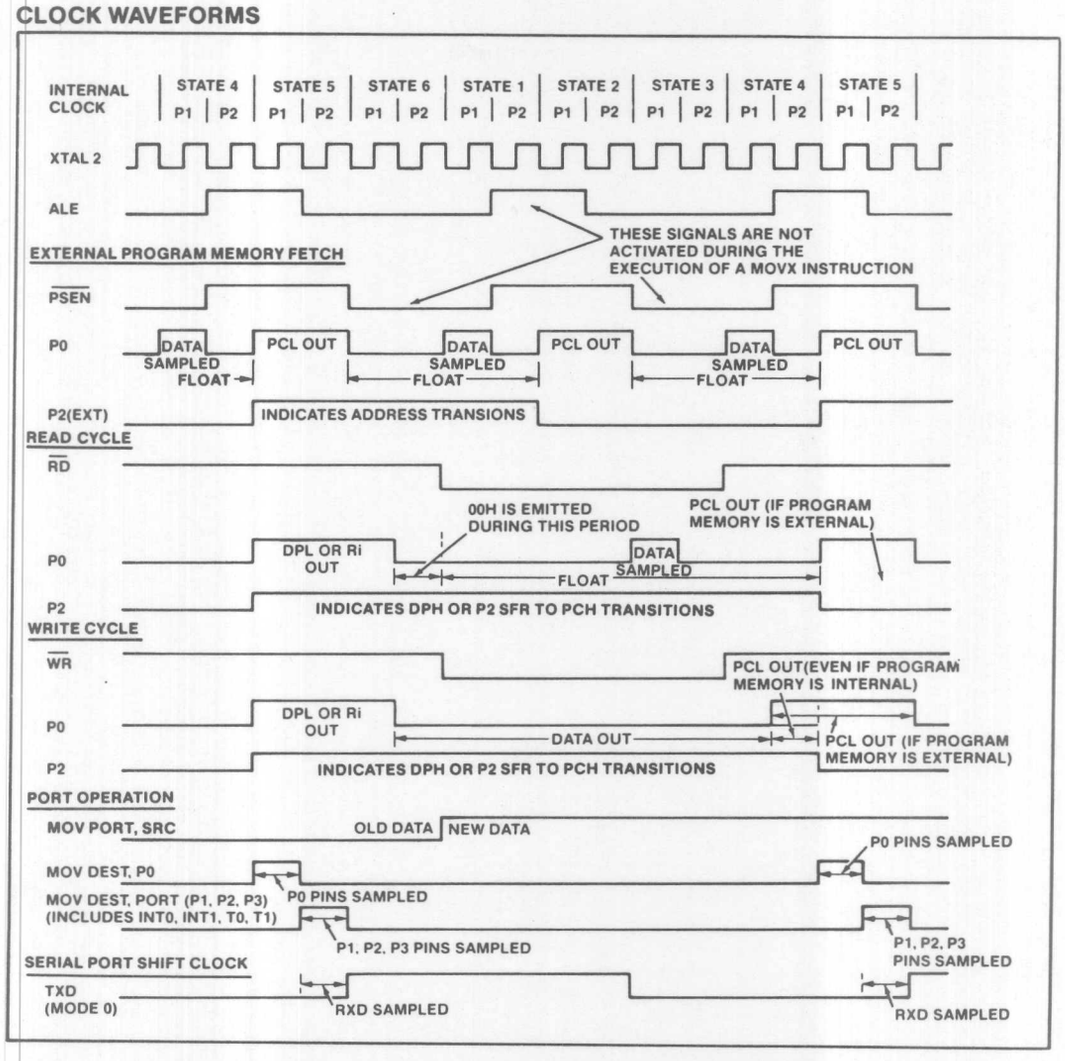
AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0."
Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

EXTERNAL CLOCK DRIVE XTAL2

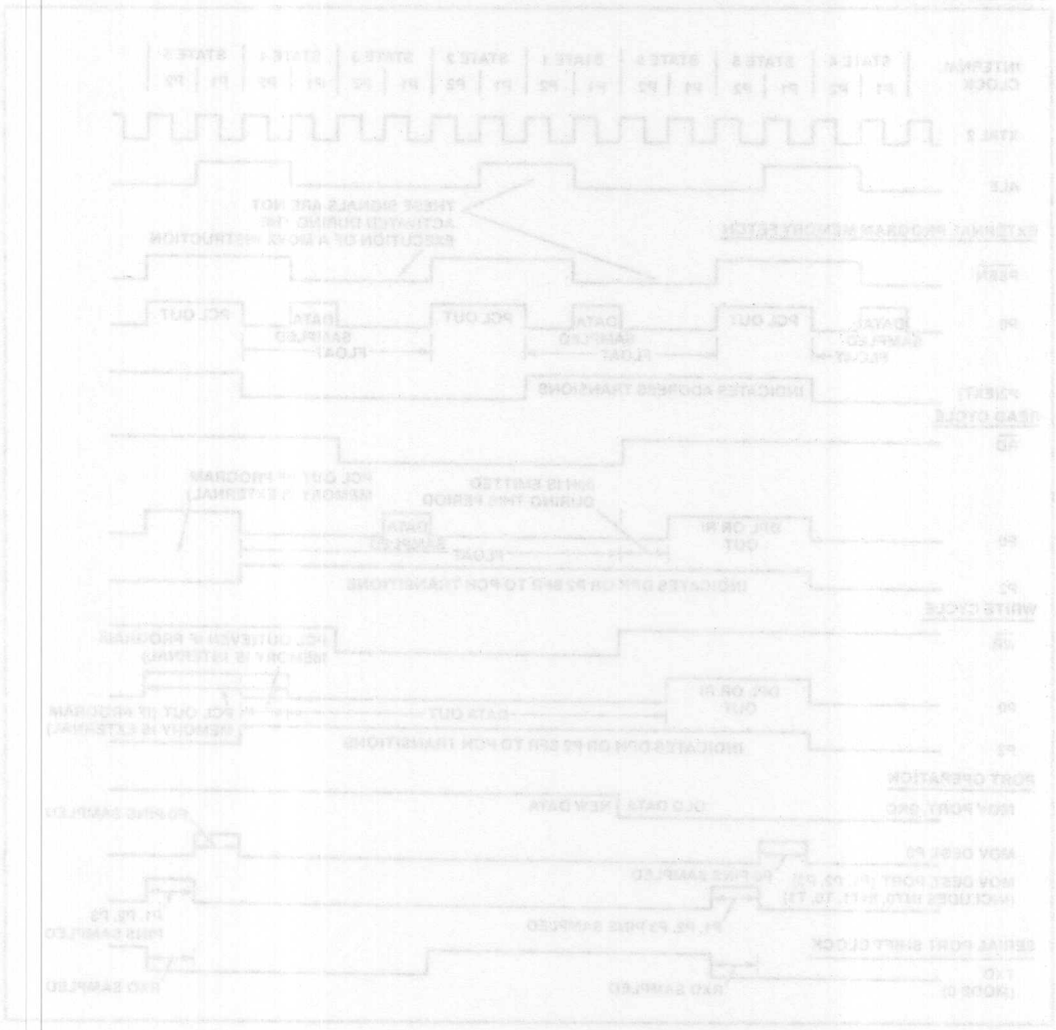


Symbol	Parameter	Variable Clock Freq = 1.2 MHz to 12 MHz		Unit
		Min	Max	
TCLCL	Oscillator Period	83.3	833.3	ns
TCHCX	High Time	20	TCLCL-TCLCX	ns
TCLCX	Low Time	20	TCLCL-TCHCX	ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though, (T_A = 25° C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.



THE BITBUS™ INTERCONNECT SERIAL CONTROL BUS SPECIFICATION

TABLE OF CONTENTS

TITLE	PAGE
1.0 GENERAL	57
1.1 SCOPE	57
1.2 DEFINITIONS	57
2.0 FUNCTIONAL OVERVIEW	58
2.1 HIERARCHICAL MODEL	58
2.2 BUS ELEMENTS	59
2.3 MODES OF OPERATION	60
2.4 LEVELS OF SPECIFICATION	63
3.0 ELECTRICAL SPECIFICATION	64
3.1 DATA ENCODING TECHNIQUES	64
3.2 DC SPECIFICATIONS	66
3.3 AC SPECIFICATIONS	71
4.0 DATA LINK PROTOCOL SPECIFICATION	75
4.1 SYSTEM STATE	75
4.2 FRAME FORMAT	76
4.3 CONTROL FIELD DEFINITION	77
4.4 DATA LINK OPERATION	80
5.0 MESSAGE PROTOCOL	85
5.1 ORDER/REPLY STRUCTURE	85
5.2 MESSAGE FORMAT	86
6.0 REMOTE ACCESS AND CONTROL	88
6.1 MESSAGE FORMAT FOR RAC	88
6.2 RAC COMMANDS	88
6.3 RAC RESPONSES	94
7.0 MECHANICAL SPECIFICATION	95
7.1 CONNECTOR SPECIFICATIONS	95
7.2 I/O BOARD SPECIFICATION	97
8.0 LEVELS OF COMPLIANCE	101
8.1 CONCEPT OF LEVEL OF COMPLIANCE	101
8.2 COMPLIANCE LEVELS	102

FIGURES

TITLE	PAGE
Figure 1a. BITBUS™ Interconnect Hierarchy	58
Figure 1b. BITBUS™ Interconnect Hierarchy	58
Figure 2. BITBUS™ Interconnect Elements	59
Figure 3. Synchronous Mode Connection	61
Figure 4. Typical Synchronous Mode Interface	61
Figure 5. Self Clocked Mode Connection	62
Figure 6. Typical Self Clocked Mode Interface	62
Figure 7. Typical Repeater	63
Figure 8. Synchronous Mode Data Encoding	65
Figure 9. Self Clocked Mode Data Encoding (NRZI)	65
Figure 10. Standard Unit Load	66
Figure 11. Transmitter Open Circuit Test Configuration	67
Figure 12. Transmitter Test Termination Configuration 1	68
Figure 13. Transmitter Test Termination Configuration 2	68
Figure 14. Transmitter Fault Conditions	69
Figure 15. Receiver Sensitivity Specification	69
Figure 16. Receiver Balance Specification	70
Figure 17. Rise and Fall Time Specification	71
Figure 18. Test Configuration for Rise and Fall Time Measurement	72
Figure 19. Transmitter Enable Timing	72
Figure 20. Data Clock Signal Pair Specification for Synchronous Mode	73
Figure 21. Data Signal Pair Specification for Synchronous Mode	73
Figure 22. Data Signal Pair Specification for Self Clocked Mode	74
Figure 23. Repeater Skew on Transitions	74
Figure 24. Repeater Skew Between Signal Pairs	74
Figure 25. Standard Frame Format	76
Figure 26. Unnumbered Control Field Format	78
Figure 27. Supervisory Control Field Format	79
Figure 28. Information Control Field Format	80
Figure 29. Slave Device State Diagram	81
Figure 30. Slave Response to Incoming Frame in NRM	82
Figure 31. Typical Initialization Sequence	83
Figure 32. Transfer Sequence Examples	84
Figure 33. Message Format	86
Figure 34. GET Function ID Reply Example	90

FIGURES (Con't)

TITLE	PAGE
Figure 35. Data Field Format for Memory Commands	91
Figure 36. Data Field Format for I/O Commands	92
Figure 37. Data Field Format for Status Commands	94
Figure 38. Printed Board Connector	96
Figure 39. Cable to Cable Connector	97
Figure 40. Standard Board Dimensions	98
Figure 41. Board Connector	99
Figure 42. Backplane Connector	99
Figure 43. Relationship of Mechanical Components	100

TABLES

TITLE	PAGE
Table 1. Number of Repeaters	75
Table 2. Unnumbered Control Fields	78
Table 3. Slave Device State Transitions and Responses	81
Table 4. Message Protocol Responses	87
Table 5. RAC Commands	89
Table 6. Function ID Code Assignments	90
Table 7. RAC Error Responses	95
Table 8. Connector Pin Assignments	96
Table 9. I/O Board Pin Assignment	101

REV.	REVISION HISTORY	PRINT DATE
A	Original Issue.	1/84

1.0 GENERAL

1.1 SCOPE

Connection of dedicated controllers and process I/O points to a control computer has been a long time problem. Traditional methods have used a standard parallel bus to interconnect I/O expansion boards, which provide the required signal conditioning. In many cases, these buses are satisfactory; however, in many other cases the characteristics of these parallel buses create significant limitations. Electrically, parallel buses limit the number of boards that can be transparently (i.e. in the same backplane) added to the system. Mechanical packaging for parallel buses often makes I/O cabling cumbersome, due to the close proximity of boards within a backplane. Most importantly, the fixed form factor and bus interface logic requirements of a parallel bus put a lower bound on future system cost reduction, regardless of the advancing capabilities of future VLSI. The BITBUS™ interconnect is a serial control bus specifically designed to address these problems.

The BITBUS interconnect allows up to 250 nodes to be easily interconnected over a physically distributed domain. Distribution capability ranges from 30 meters for the synchronous mode of operation to thousands of meters for the self clocked mode of operation. The different modes of operation are optimized for a wide range of applications covering the spectrum from high speed servo motor control in robotics, to long distance environmental control. In all cases the BITBUS interconnect is optimized for the high speed transfer of short control messages in a hierarchical system.

The level of specification for the BITBUS interconnect is somewhat different than for traditional parallel bus structures. In the past, serial connections have been slow, software intensive and hard to use. A goal of the BITBUS interconnect is to provide an easy to use, high performance serial interconnect that is transparent to the application programmer. For this purpose, this specification not only defines electrical and data link protocol aspects of the bus, as is typical for parallel buses, but also specifies a message structure and protocol for a multitasking environment, and a set of high level commands for remote I/O access and application task control. This allows standard high level software interfaces to be written, off-loading the application programmer of this complication. More importantly, this high level of specification allows the standard interface to be driven into silicon, further reducing the interface overhead, cost and complexity.

This specification has been prepared for those users intending to design or evaluate products that will be compatible with the BITBUS interconnect. The intent of this specification is to specify the interface requirements only. The implementation of interface logic and software is left to the user.

1.2 DEFINITIONS

System — A set of interconnected elements which achieve a given objective through performing a specific function. In this specification a BITBUS system consists of one master node and from one to 250 slave nodes attached to the same BITBUS interconnect.

Message-Passing — The transfer and control of structured data between two tasks.

Task — An entity which competes for system resources in order to execute a program.

Protocol — The rules by which information is exchanged across an interface.

Operation — The process whereby information is transferred between two elements across an interface.

Interface — A shared boundary between two elements within a system.

Bit-Cell-Time — The time interval required to transfer one bit of data on a serial line.

2.0 FUNCTIONAL OVERVIEW

The BITBUS interconnect concept allows highly reliable, low cost distributed expansion of process I/O points and intelligent I/O controllers. This section provides an overall understanding of the concept by describing the hierarchical model, bus elements, modes of operation and levels of specification. Detailed specifications are provided in subsequent sections.

2.1 HIERARCHICAL MODEL

The BITBUS interconnect is defined to provide a high speed serial control bus for hierarchical systems. This model, shown in figure 1, implies that a single master must communicate with multiple slaves. These slaves may consist of simple I/O points or they may be highly intelligent controllers. By adopting this architectural model, the BITBUS interconnect is able to provide a highly capable, low cost solution, especially for those applications that match the model. This is achieved through a significant reduction in protocol complexity and overhead that would otherwise be necessary to implement a more complex multiple access type protocol. Evidence of this simplicity can be seen by comparing the level of protocol support provided in hardware for the BITBUS interconnect as compared to other more complex protocols.

The hierarchical model for the BITBUS interconnect may consist of one or multiple levels as shown in figures 1a and 1b respectively. The multiple level hierarchy is extremely useful in many applications, especially where BITBUS interconnects must run at different speeds. This specification provides the necessary hooks to implement such a structure with minimal overhead. In this specification a BITBUS system refers to only the single level hierarchy. The multiple level hierarchy consists of multiple BITBUS systems.

The goal of the BITBUS interconnect is to provide a message passing interface between tasks at the master node and tasks at multiple slave nodes within the hierarchical model. This is accomplished through an order/reply message protocol. Tasks on the master node issue orders to tasks on the slave nodes and tasks on the slave nodes respond with replies. This level of support effectively hides the serial nature of the BITBUS interconnect from the application programmer, resulting in a relatively simple interface.

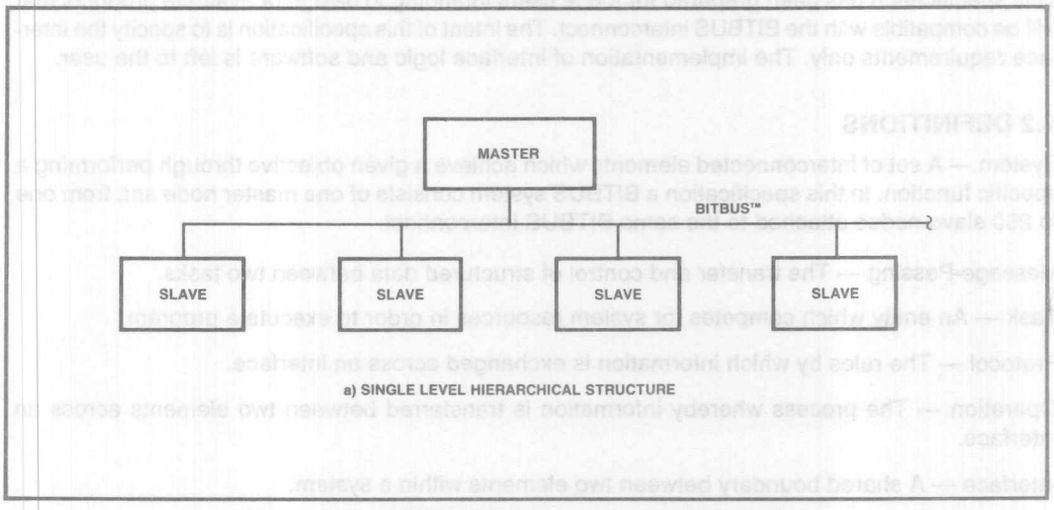


Figure 1a. BITBUS™ Interconnect Hierarchy

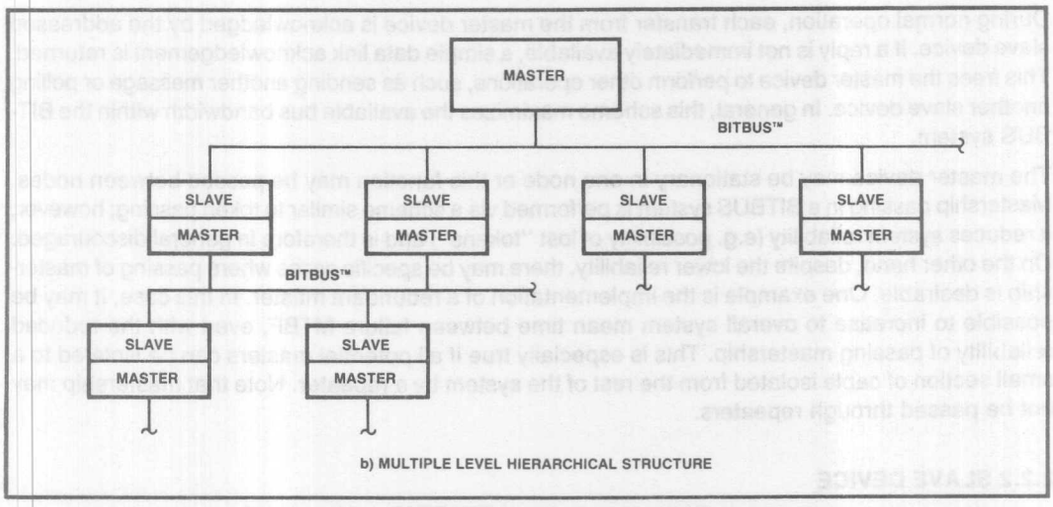


Figure 1b. BITBUS™ Interconnect Hierarchy

2.2 BUS ELEMENTS

The basic element in a BITBUS system is a node. A node may consist of either a device, a device and an extension or a repeater. There are two types of devices: the master device and a slave device. This section explains the unique elements of the BITBUS system: the master device, the slave device, the extension and the repeater. Figure 2 illustrates these elements.

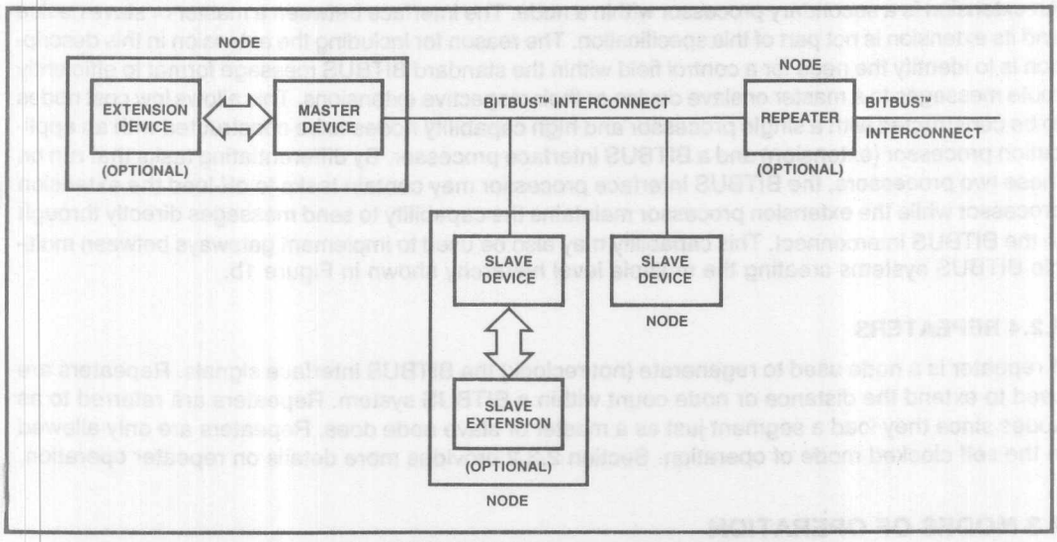


Figure 2. BITBUS™ Interconnect Elements

2.2.1 MASTER DEVICE

The master device controls all BITBUS interconnect operations. There is only one master device per BITBUS system. The master device initiates a transaction by sending an order message to a slave device. It then continues to poll the slave device until a reply message is returned.

During normal operation, each transfer from the master device is acknowledged by the addressed slave device. If a reply is not immediately available, a simple data link acknowledgement is returned. This frees the master device to perform other operations, such as sending another message or polling another slave device. In general, this scheme maximizes the available bus bandwidth within the BITBUS system.

The master device may be stationary in one node or this function may be passed between nodes. Mastership passing in a BITBUS system is performed via a scheme similar to token passing; however, it reduces system reliability (e.g. possibility of lost "tokens") and is therefore in general discouraged. On the other hand, despite the lower reliability, there may be specific cases where passing of mastership is desirable. One example is the implementation of a redundant master. In this case, it may be possible to increase to overall system mean time between failure MTBF, even with the reduced reliability of passing mastership. This is especially true if all potential masters can be isolated to a small section of cable isolated from the rest of the system by a repeater. Note that mastership may not be passed through repeaters.

2.2.2 SLAVE DEVICE

A slave device is a replier in a BITBUS system. There may be up to 250 slaves per system (addresses 1-250). As a replier, the slave may not spontaneously initiate a transfer. Rather, it may only respond to orders from the master device with replies. This approach significantly simplifies the slave device protocol requirements allowing powerful, low cost solutions to be implemented using highly integrated microcontrollers.

2.2.3 EXTENSION

An extension is a secondary processor within a node. The interface between a master or slave device and its extension is not part of this specification. The reason for including the extension in this description is to identify the need for a control field within the standard BITBUS message format to efficiently route messages to a master or slave device or their respective extensions. This allows low cost nodes to be constructed with a single processor and high capability nodes to be constructed with an application processor (extension) and a BITBUS interface processor. By differentiating tasks that run on these two processors, the BITBUS interface processor may contain tasks to off-load the extension processor while the extension processor maintains the capability to send messages directly through to the BITBUS interconnect. This capability may also be used to implement gateways between multiple BITBUS systems creating the multiple level hierarchy shown in Figure 1b.

2.2.4 REPEATERS

A repeater is a node used to regenerate (not reclock) the BITBUS interface signals. Repeaters are used to extend the distance or node count within a BITBUS system. Repeaters are referred to as nodes since they load a segment just as a master or slave node does. Repeaters are only allowed in the self clocked mode of operation. Section 2.3.2 provides more details on repeater operation.

2.3 MODES OF OPERATION

The BITBUS interconnect may operate in one of two modes: synchronous mode and self clocked mode. The purpose for multiple modes is to provide a wide range of performance/distance options for a variety of applications. In this section each mode is presented with its corresponding features and signal line definitions. Note that in both modes of operation all BITBUS interconnect signals are differential. The terminology used to specify the signal pairs is NAME and NAME*. State definition for the signal pairs is provided in section 3.1.1 of this specification.

2.3.1 SYNCHRONOUS MODE

The synchronous mode of operation is optimized for the highest performance over a relatively short distance. This mode can interconnect up to 28 nodes over 30 meters at transmission speeds between 500 kbits/sec and 2.4 mbits/sec. A typical system is shown in figure 3.

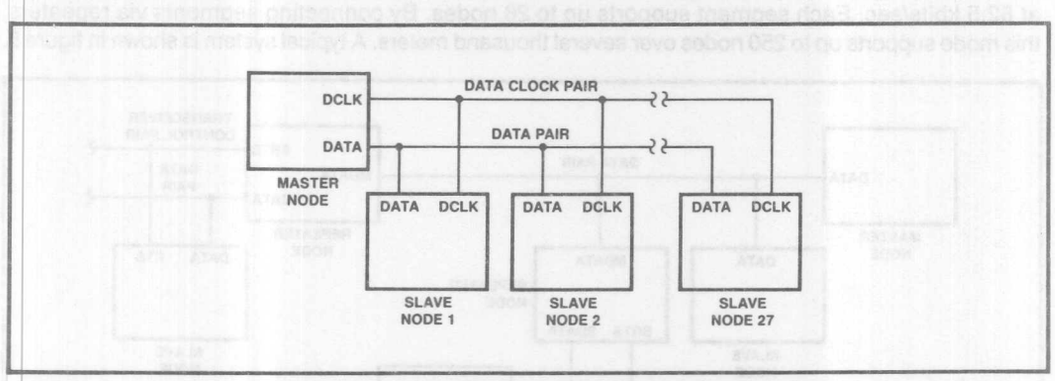


Figure 3. Synchronous Mode Connection

The synchronous mode uses two differential signal pairs: one for data (DATA, DATA*) and one for the data clock (DCLK, DCLK*). The data signal pair carries data which is referenced to the data clock signal pair. Data changes on the "falling edge" of the data clock pair and is sampled on the "rising edge". A sample interface circuit is shown in figure 4. Note that the data clock source is always at the transmitting node.

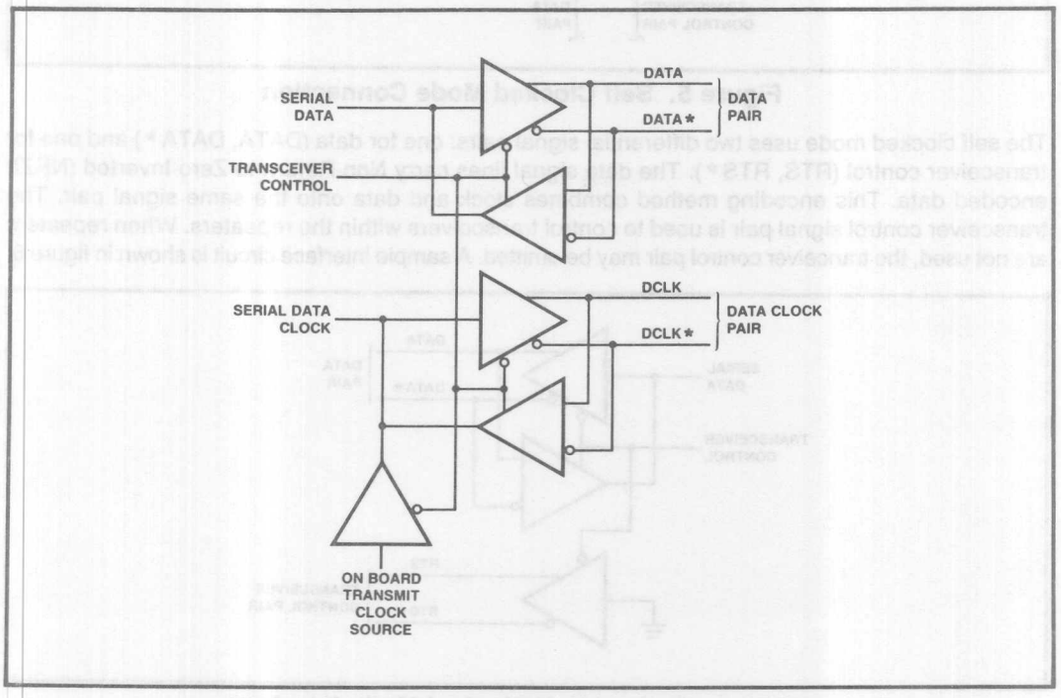


Figure 4. Typical Synchronous Mode Interface

2.3.2 SELF CLOCKED MODE

The self clocked mode of operation is provided to allow longer distance operation. There are two standard bit rates for this mode of operation: 375 kbits/sec and 62.5 kbits/sec. The BITBUS interconnect supports segments up to 300 meters in length at 375 kbits/sec and 1200 meters in length at 62.5 kbits/sec. Each segment supports up to 28 nodes. By connecting segments via repeaters, this mode supports up to 250 nodes over several thousand meters. A typical system is shown in figure 5.

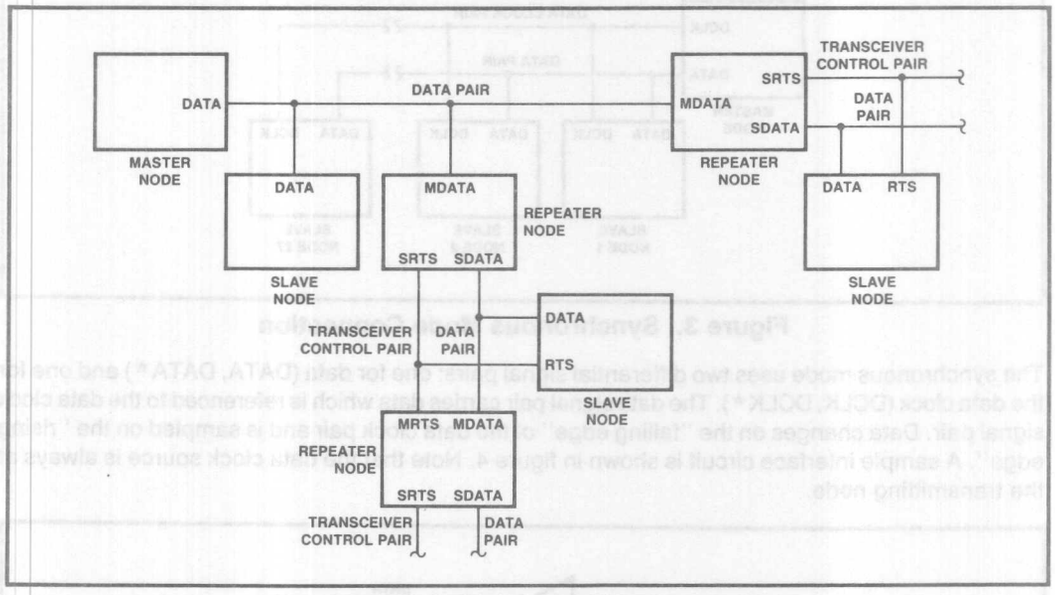


Figure 5. Self Clocked Mode Connection

The self clocked mode uses two differential signal pairs: one for data (DATA, DATA*) and one for transceiver control (RTS, RTS*). The data signal lines carry Non-Return to Zero Inverted (NRZI) encoded data. This encoding method combines clock and data onto the same signal pair. The transceiver control signal pair is used to control transceivers within the repeaters. When repeaters are not used, the tranceiver control pair may be omitted. A sample interface circuit is shown in figure 6.

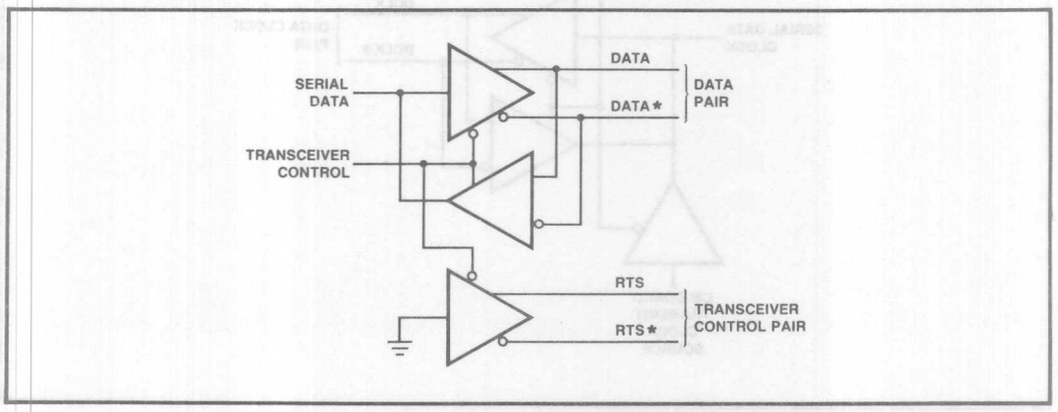


Figure 6. Typical Self Clocked Mode Interface

A BITBUS repeater may or may not provide electrical isolation, depending on the application requirements. In either case, when a slave device is not transmitting, biasing resistors enable the repeater tranceivers away from the master device, allowing the master device to transmit to all slaves. When a slave device responds, it reverses the polarity of the transceiver control pair, which reverses the direction of all repeaters between it and the master device. Figure 7 shows a sample repeater circuit.

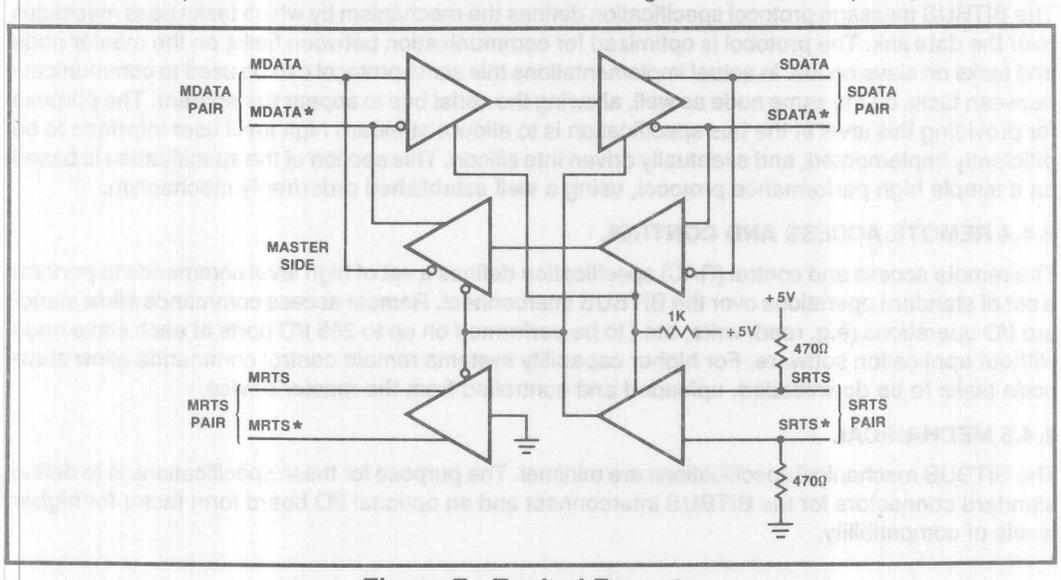


Figure 7. Typical Repeater

2.4 LEVELS OF SPECIFICATION

The BITBUS interconnect is designed to provide a highly capable, easy to use interface to the application programmer. In order to be easy to use, this interface must hide the complexity of controlling the serial link from the user through a standard hardware/software interface. In order to be highly capable, the BITBUS interconnect must be defined at a high enough level to allow this hardware and software interface to be efficient, and to a large degree be driven into hardware. These requirements result in the need for the following levels of specification:

Electrical Interface
Data Link Protocol
Message Protocol
Remote Access and Control
Mechanical

2.4.1 ELECTRICAL INTERFACE

The BITBUS electrical interface specification defines the mechanisms by which bits are transferred. This section of the specification is simplified by the fact that the BITBUS interconnect consists of only two signal pairs and the electrical interface is based to a large degree on existing standards. This level of the specification is optimized for low cost implementations (i.e. low cost cable and low cost transceivers).

2.4.2 DATA LINK PROTOCOL

The BITBUS data link protocol specification defines the mechanisms by which packets are reliably transferred across the electrical interface. Reliability aspects include error detection, and in many

cases error recovery. This is important to the BITBUS interconnect definition since it makes the application less sensitive to the serial nature of the interconnect. This section of the specification is also based to a large degree on existing standards.

2.4.3 MESSAGE PROTOCOL

The BITBUS message protocol specification defines the mechanism by which tasks pass messages over the data link. The protocol is optimized for communication between tasks on the master node and tasks on slave nodes. In actual implementations this same protocol can be used to communicate between tasks on the same node as well, allowing the serial bus to appear transparent. The purpose for providing this level in the bus specification is to allow a standard high level user interface to be efficiently implemented, and eventually driven into silicon. This section of the specification is based on a simple high performance protocol, using a well established order/reply mechanism.

2.4.4 REMOTE ACCESS AND CONTROL

The remote access and control (RAC) specification defines a set of high level commands to perform a set of standard operations over the BITBUS interconnect. Remote access commands allow standard I/O operations (e.g. read, write, etc.) to be performed on up to 256 I/O ports at each slave node without application software. For higher capability systems remote control commands allow slave node tasks to be downloaded, uploaded and controlled from the master device.

2.4.5 MECHANICAL

The BITBUS mechanical specifications are minimal. The purpose for these specifications is to define standard connectors for the BITBUS interconnect and an optional I/O board form factor for higher levels of compatibility.

3.0 ELECTRICAL SPECIFICATION

The BITBUS interconnect electrical specification requires definition of data encoding techniques and DC/AC parameters for the interface logic and interconnect cable. This section provides the required definition and specifically points out where existing standards are used as a basis for the specification.

3.1 DATA ENCODING TECHNIQUES

The BITBUS interconnect uses different data encoding techniques for the two modes of operation. Each is discussed separately below, along with a general definition of signal pair state.

3.1.1 GENERAL SIGNAL PAIR STATE DEFINITION

All signals on the BITBUS interconnect are differential pairs. The naming convention for these pairs identifies the two signal lines as NAME and NAME*. An active signal (logical 1) is present when the NAME signal line is at a higher electrical potential than the NAME* signal line. An inactive signal (logical 0) is present when the NAME signal line is at a lower electrical potential than the NAME* signal line.

3.1.2 SYNCHRONOUS MODE

Data encoding is relatively simple in the synchronous mode. The logical value of each bit is simply indicated by the level of the data signal pair at the bit cell center. The bit cell center is defined as the rising edge of the data clock pair. A logical zero is represented by the DATA signal being at a lower electrical potential than the DATA* signal at the bit cell center. A logical one is represented by the DATA signal being at a higher electrical potential than the DATA* signal at the bit cell center. These relationships are shown in figure 8. Data encoding for the synchronous mode of operation

must also provide zero bit insertion/deletion. This is required to support the frame delimiting flags defined in section 4 of this specification. The flags are defined as the bit pattern 01111110. In order to guarantee uniqueness of this bit pattern, there may be no more than five consecutive ones in the bit stream (other than flags). This is accomplished using zero bit insertion (by the transmitter) and deletion (by the receiver). Specifically, the transmitter inserts a zero into the bit stream anytime it detects five consecutive ones (except for flags) regardless of the next bit value. Receivers then remove any zero from the bit stream that is preceded by five ones. The receiver must also detect the 01111110 bit pattern as a frame delimiting flag.

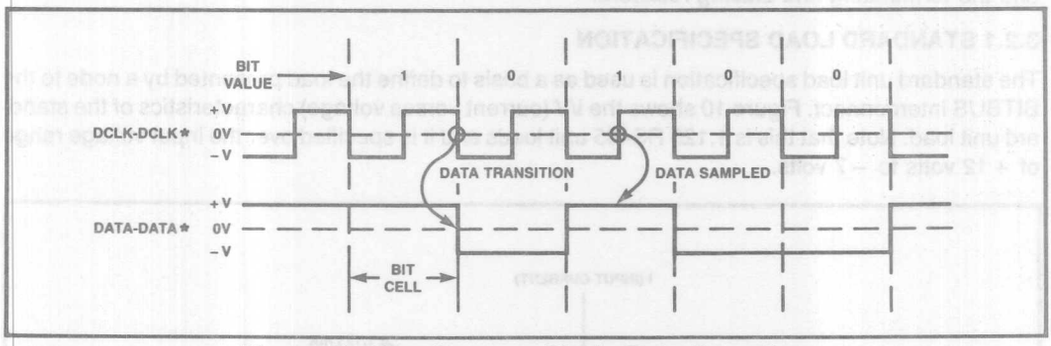


Figure 8. Synchronous Mode Data Encoding

3.1.3 SELF CLOCKED MODE

The self clocked mode of operation uses standard NRZI encoding with zero bit insertion/deletion. This encoding method combines serial data and serial clock information onto a single signal pair (DATA, DATA*). A logical zero is represented by a change in the polarity of the data signal pair from the proceeding bit cell (i.e. a transition occurs on the bit cell boundary). A logical one is represented by no change in the polarity of the data signal pair from the preceding bit cell. These relationships are shown in figure 9. As in the synchronous mode, zero bit insertion/deletion is provided to guarantee uniqueness of the frame delimiting flag pattern.

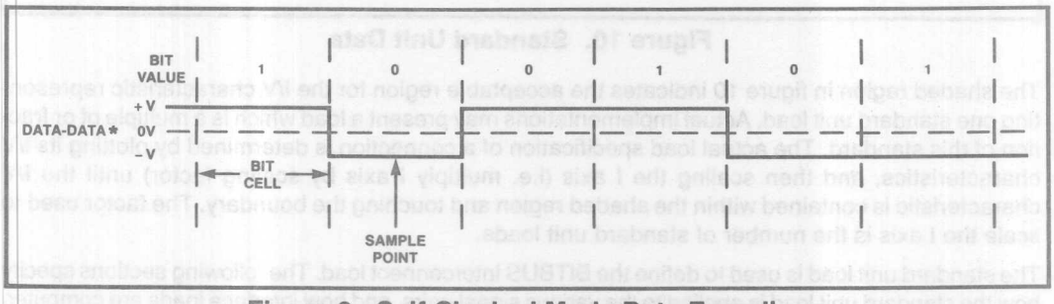


Figure 9. Self Clocked Mode Data Encoding

The self clocked mode of operation requires that receivers recover the serial clock from the data bit stream. The BITBUS interconnect is defined with the assumption that a digital phase locked loop (DPLL) with a 16× reference clock will be used for this purpose. For proper operation, the DPLL must be synchronized (i.e. be in phase with the transmitter clock) before, and remain synchronized during the transmission of a frame. Initial synchronization of the receivers DPLL is guaranteed by the transmission of a preframe sync (PFS) prior to the frame. The PFS consists of a minimum of eight zeros (i.e. transitions) that allow the DPLL to adjust (one reference clock at a time) until synchronized.

Once synchronized, the DPLL uses the transitions (i.e. zeros) within the bit stream to make fine adjustments of ±1 reference clock cycle. Note that zero bit insertion/deletion guarantees a transition at least every seven bit cells, allowing a reasonable tolerance on the reference clock.

3.2 DC SPECIFICATIONS

The DC specifications for the BITBUS interconnect are based on the RS485 electrical standard. This section defines the characteristics of a standard load, a transmitter, a receiver, the interconnect cable and the terminating and biasing resistors.

3.2.1 STANDARD LOAD SPECIFICATION

The standard unit load specification is used as a basis to define the load presented by a node to the BITBUS interconnect. Figure 10 shows the I/V (current verses voltage) characteristics of the standard unit load. Note that this is 1.125 RS485 unit loads and it is specified over the input voltage range of +12 volts to −7 volts.

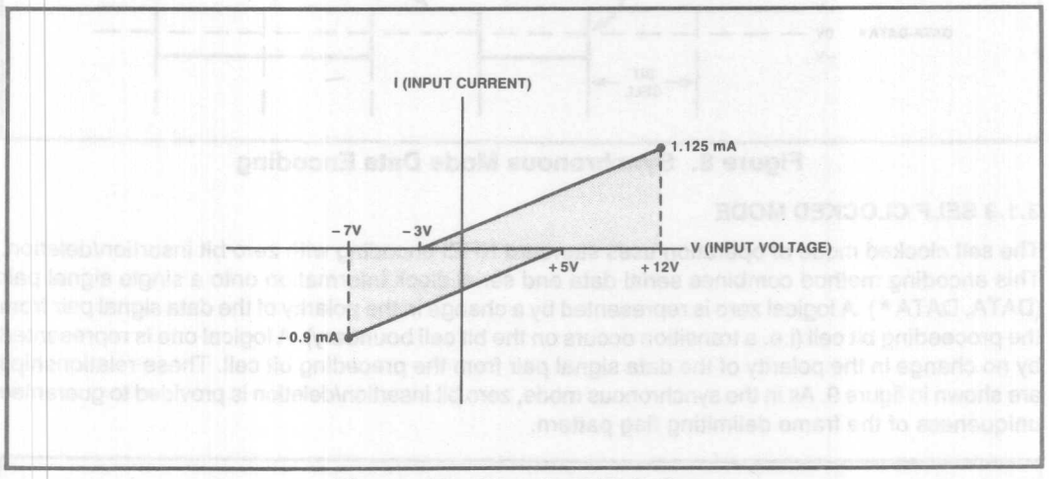


Figure 10. Standard Unit Data

The shaded region in figure 10 indicates the acceptable region for the I/V characteristic representing one standard unit load. Actual implementations may present a load which is a multiple of or fraction of this standard. The actual load specification of a connection is determined by plotting its I/V characteristics, and then scaling the I axis (i.e. multiply I axis by scaling factor) until the I/V characteristic is contained within the shaded region and touching the boundary. The factor used to scale the I axis is the number of standard unit loads.

The standard unit load is used to define the BITBUS interconnect load. The following sections specify how the standard unit load is applied to the various signal pairs, and how interface loads are computed for the synchronous and self clocked modes of operation.

3.2.1.1 Data And Data Clock Pair Specification

The load for the data pair (DATA, DATA*) and data clock pair (DCLK, DCLK*) consist of a disabled transmitter and a receiver as shown in figures 4, 6 and 7. The DC characteristics of these loads are specified as a multiple of the standard unit load shown in figure 10. The standard unit load specification has been defined such that most implementations with standard off the shelf components will be one standard unit load.

3.2.1.2 Transceiver Control Pair Specification

The transceiver control pair (RTS, RTS*) is not symmetrical (i.e. there is not a transmitter and receiver at each node) like the data and data clock pairs. Rather, this pair has one "receiver only" node (in the repeater as shown in figure 7) and multiple "transmitter only" nodes (as shown in figures 6 and 7). The DC characteristics of the transmitter only loads are specified as a multiple of 1/9 the standard unit load shown in figure 10. For example, if a node presents 2/9 standard unit loads on this pair it is counted as 2 BITBUS interconnect loads. The DC characteristics of the receiver only node shall not exceed 25 standard unit loads, including the biasing resistors specified in section 3.2.5. With this definition, 27 receiver only nodes and one transmitter only node are equivalent to 28 standard unit loads ($27 \times 1/9 + 25 = 28$).

3.2.1.3 Synchronous Mode Load

The load presented to the BITBUS interconnect in synchronous mode is defined by determining the number of standard unit loads for each signal pair. The largest number is the specification. For example if the data signal pair presents a load equal to the standard unit load, and the data clock signal pair presents a load equal to 2 standard unit loads, the interface would be specified as 2 loads.

3.2.1.4 Self Clocked Mode Load

The load presented to the BITBUS interconnect in the self clocked mode is also defined by determining the number of standard unit loads for each signal pair. For example if the data signal pair presents a load equal to the standard unit load, and the transceiver control pair presents a load equal to 2/9 standard unit loads, the interface would be specified as 2 loads. If the tranceiver control pair were not used (i.e. no repeater connections) the interface would be specified as 1 load.

3.2.2 TRANSMITTER SPECIFICATION

The standard BITBUS transmitter shall meet the requirements of an RS485 generator. It shall be capable of driving a 60 ohm termination (120 ohms at each end of the cable) and up to 32 RS485 unit loads (28 standard unit loads in Figure 10). For reference, the DC requirements are repeated in this specification. For the figures in this section all output voltages apply to both logic states and the symbol Δ in front of a voltage refers to the difference between that voltage in the two logic states.

3.2.2.1 Open Circuit Specification

The transmitter open circuit test configuration is shown in figure 11. All relevant test parameters are shown in the figure. This test is used to verify the output voltage characteristics when no load is applied.

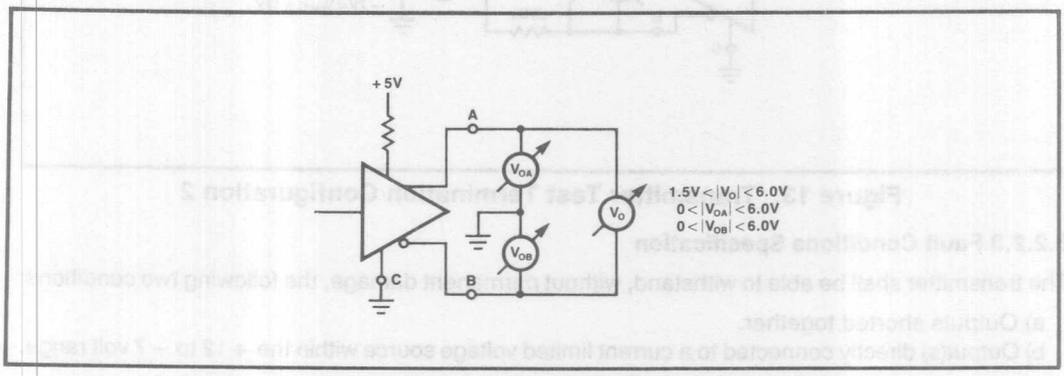


Figure 11. Transmitter Open Circuit Test Configuration

3.2.2.2 Test Termination Specification

The transmitter shall be able to drive the two test configurations shown in figures 12 and 13. All relevant test parameters are shown in the figures.

In figure 12 the transmitter drives a 54 ohm test load consisting of two 27 ohm resistors in series. This test termination is used to verify the output drive (V_O) capability and the signal pair offset voltage (V_{OS}) characteristics.

In figure 13 the transmitter drives a 60 ohm termination plus a load that simulates 32 RS485 unit loads through the RS485 common mode voltage range of +12 to −7 volts. This test termination is used to verify output drive capability under worst case command mode conditions.

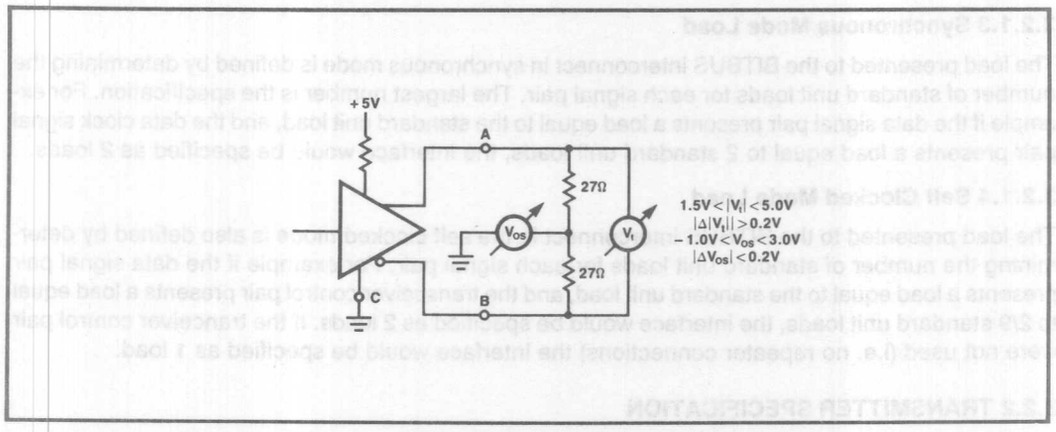


Figure 12. Transmitter Test Termination Configuration 1

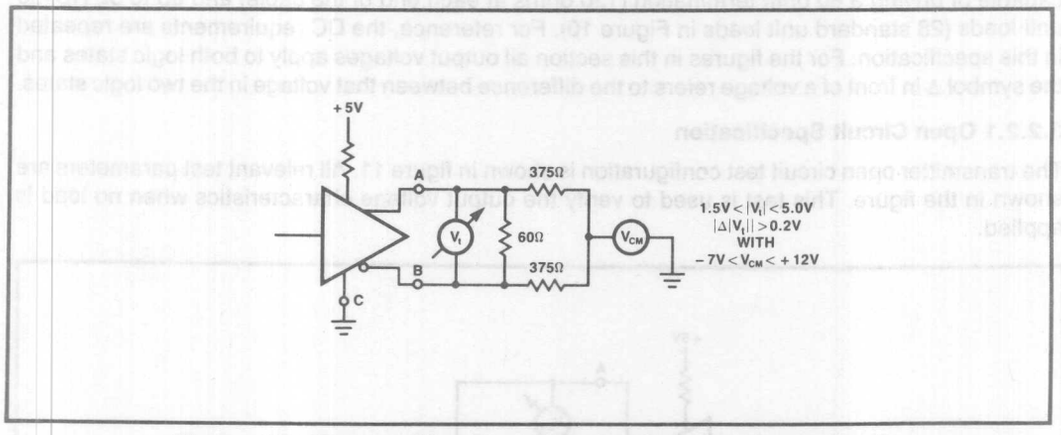


Figure 13. Transmitter Test Termination Configuration 2

3.2.2.3 Fault Conditions Specification

The transmitter shall be able to withstand, without permanent damage, the following two conditions:

a) Outputs shorted together.

b) Output(s) directly connected to a current limited voltage source within the +12 to −7 volt range.

Figure 14 shows the test configurations.

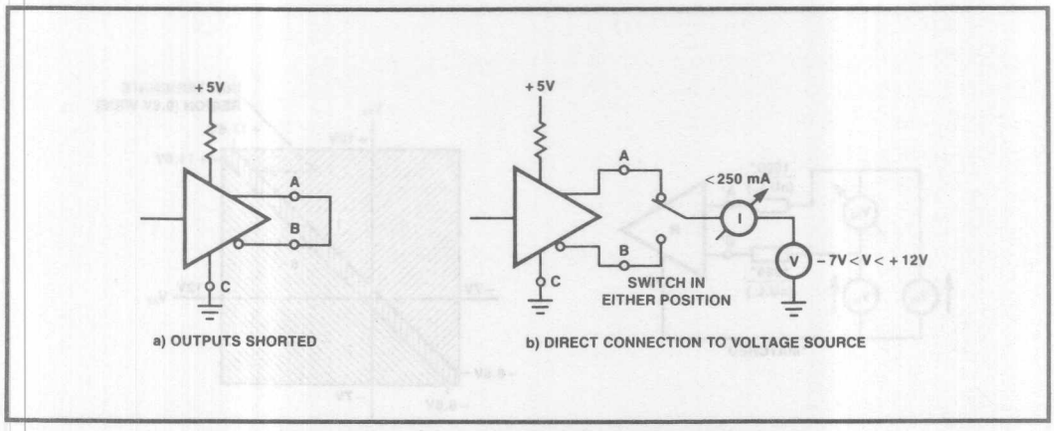


Figure 14. Transmitter Fault Conditions

3.2.3 RECEIVER SPECIFICATION

The standard BITBUS receiver shall meet the requirements of an RS485 receiver. Receiver load considerations are included in section 3.2.1. For reference, the input sensitivity and balance requirements are repeated in this section.

3.2.3.1 Sensitivity Specification

The receiver sensitivity specification is shown in figure 15. The allowable range of input voltages appearing at the receiver inputs, referenced to the receiver ground, shall be between +12 and −7 volts. Within this range a differential voltage of 200 millivolts or more shall be detectable as a valid logic state. The states are indicated in figure 15.

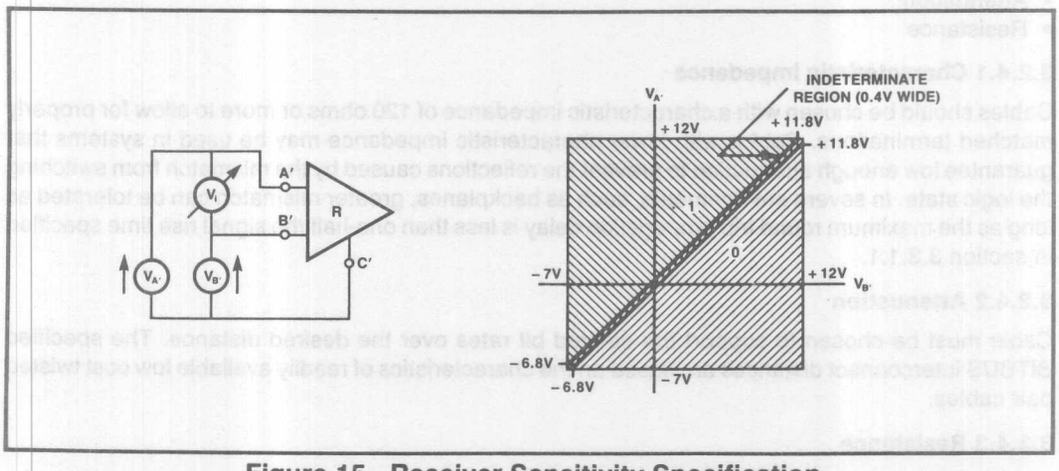


Figure 15. Receiver Sensitivity Specification

3.2.3.2 Balance Specification

The receiver balance specification is shown in figure 16. For this test the input resistor shown is 1500 ohms divided by the number of RS485 unit loads of the receiver. Within the common mode input range of +12 to −7 volts, a differential input of 400 millivolts or more shall be detectable as a valid logic state. The states are indicated in figure 16.

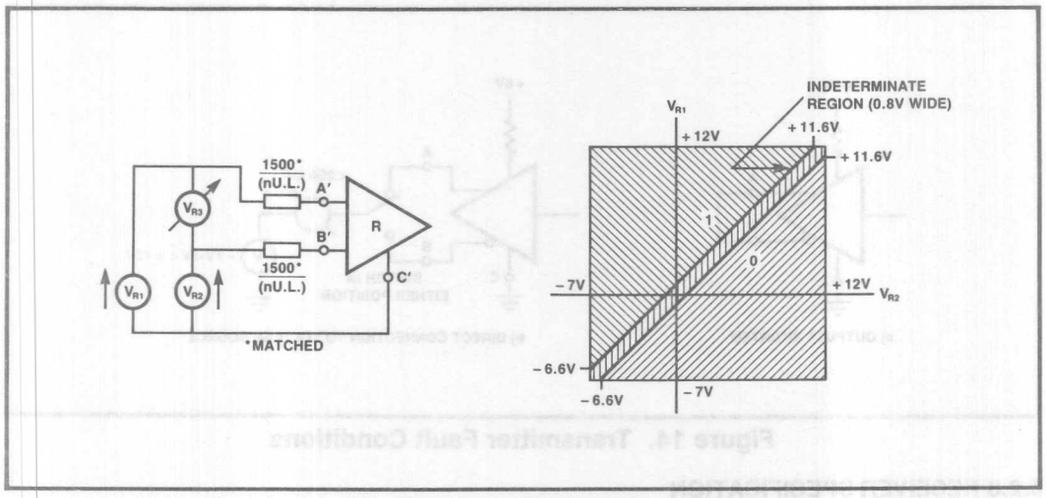


Figure 16. Receiver Balance Specification

3.2.4 CABLE CHARACTERISTICS

The BITBUS is defined to operate on a variety of cables including a low cost twisted pair (shielded or unshielded depending on application requirements) or backplane traces. Specific cable characteristics are not rigidly specified in order to allow flexibility for various implementations (i.e. only the interfaces are rigidly specified). When selecting a cable, the following aspects must be considered. Generally these parameters can be found in the cable data sheet.

- Characteristic Impedance
- Attenuation
- Resistance

3.2.4.1 Characteristic Impedance

Cables should be chosen with a characteristic impedance of 120 ohms or more to allow for properly matched terminations. Cables with lower characteristic impedance may be used in systems that guarantee low enough attenuation to prevent the reflections caused by the mismatch from switching the logic state. In severe environments, such as backplanes, greater mismatch can be tolerated as long as the maximum round trip propagation delay is less than one-half the signal rise time specified in section 3.3.1.1.

3.2.4.2 Attenuation

Cable must be chosen to support the desired bit rates over the desired distance. The specified BITBUS interconnect distances are based on the characteristics of readily available low cost twisted pair cables.

3.2.4.3 Resistance

Cable must be chosen with a low enough DC resistance to guarantee sufficient voltage across the termination to be detected by receivers. This parameter, the attenuation and the magnitude of reflections need to be considered together to determine the worst case noise margin at the receivers.

3.2.5 TERMINATION AND BIASING

The BITBUS interconnect requires the cable to be terminated for proper operation. There are two cases to be considered: the signal pair termination and the transceiver control pair biasing.

3.2.5.1 Termination

All BITBUS interconnect cables must be terminated at both ends for proper operation. The terminations shall be located at the extreme ends of the cable. The value of each termination shall be 120 ohms or greater and should be chosen to match the characteristic impedance of the cable as closely as possible.

3.2.5.2 Transceiver Control Biasing

In addition to terminations, the transceiver control pair shall have biasing resistors at its receiver only node. These resistors shall be 470 ohms ±5%, one being connected to +5 volts ±5% and the other connected to ground as shown in figure 7. The receiver only node may be located anywhere on the cable segment.

3.3 AC SPECIFICATIONS

The AC specifications for the BITBUS interconnect requires definition of the signal line characteristics, transmitter enable timing, self clocked mode timing, and repeater timing.

3.3.1 SIGNAL LINE CHARACTERISTICS

The BITBUS signal lines must maintain a reasonable level of signal integrity to guarantee proper operation. Specifically, there needs to be bounded rise and fall times and reflection guidelines.

3.3.1.1 Rise And Fall Time Specification

The BITBUS signal lines shall have rise and fall times between 25 and 100ns as shown in figure 17. This measurement shall be made in the test configuration shown in figure 18.

In actual systems, the rise and fall times may be slower than the above specification as long as the following two conditions are met.

a) The rise and fall time of any node shall meet the above specification (Figure 17) into the test load (Figure 18).
b) The rise and fall times in the actual system shall not be less than 0.3 times the bit cell width, measured anywhere in the system.

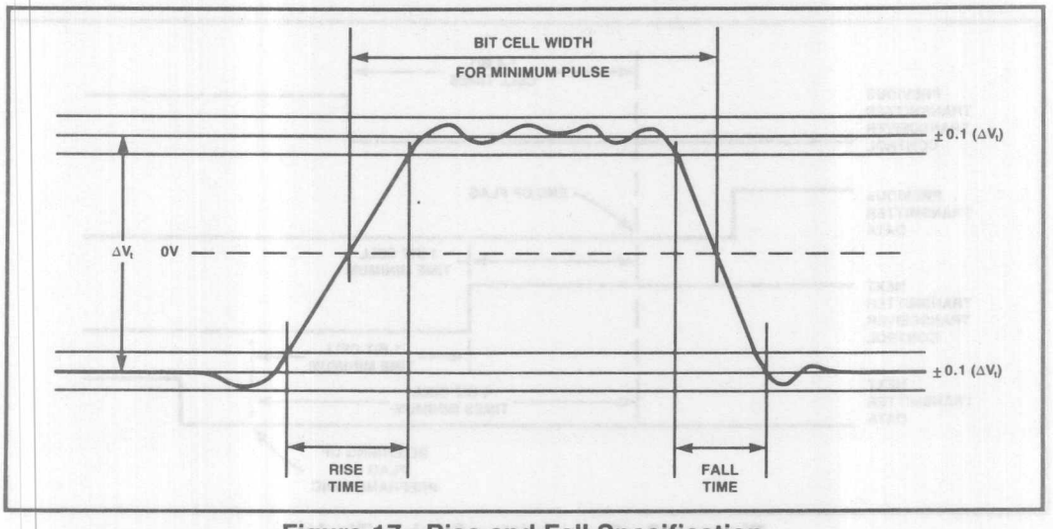


Figure 17. Rise and Fall Specification

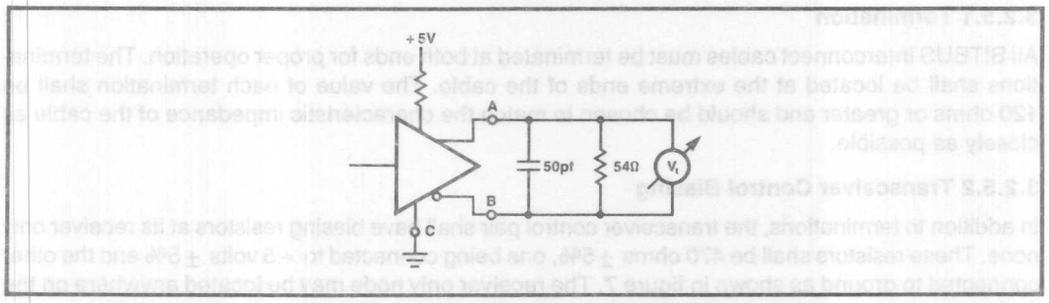


Figure 18. Test Configuration for Rise and Fall Time Measurement

3.3.1.2 Reflection Guidelines

Transitions on the BITBUS signal lines shall meet the requirements shown in figure 17 when driving the test load in figure 18. The transition from 10% to 90% of the final steady state value shall be monotonic. Ringing shall be limited to ±10% around the final steady state value.

RS485 is a robust standard, providing a significant DC noise margin. In practical systems where reflections may be unavoidable these noise margins shall be used to guarantee proper operation. Potential reflections (due to lumped capacitive loads, mismatched terminations, or excessively long stubs) should be considered together with the DC losses specified in section 3.2.4 when choosing cable. In all cases, the voltage at any node shall remain in the +12 volt to −7 volt common mode range except for pulses of less than 15 microseconds, at less than 1% duty cycle, which shall be bounded to ±25 volts.

3.3.2 TRANSMITTER ENABLE TIMING

The BITBUS interconnect is designed to operate without transceiver control signals (repeaters accepted). In order to guarantee proper operation, this requires specifications for transmitter turn on and turn off in order to prevent a "dribbling" (enabled after end of frame) transmitter from corrupting the data of another transmitter. Figure 19 summarizes these parameters.

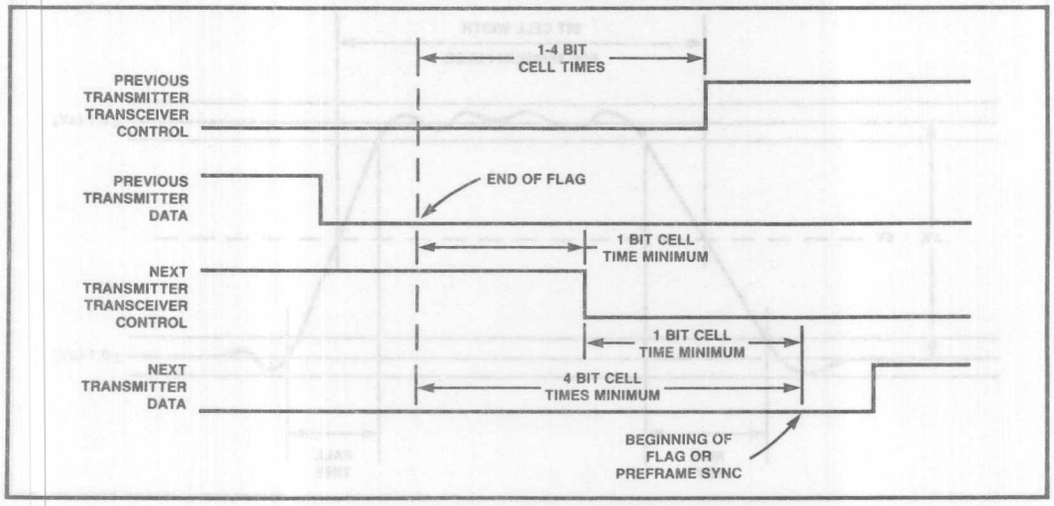


Figure 19. Transmitter Enable Timing

Transmitters shall be enabled a least one bit cell time prior to valid data transmission (opening flag in synchronous mode, preframe synch in self clocked mode). A transmitter shall not be enabled until at least one bit cell time after detecting the closing flag of a previous frame. Finally a transmitter shall guarantee that the first valid bit of data is not transmitted until any previous transmitter has been disabled.

Transmitters shall be disabled between one and four bit cell times after the end of the closing flag. Note that this specification leads to a potential three bit cell time contention period during turn around. This is acceptable based on the fault condition specification of section 3.2.2.3.

3.3.3 SYNCHRONOUS MODE TIMING

The synchronous mode of operation requires specification of the data clock signal pair timing and the data pair signal timing.

The data clock signal pair timing is shown in figure 20. All receivers shall be able to receive at speeds up to 2.4 Mbits/sec. Transmitters may transmit between 500 kbits/sec and 2.4 Mbits/sec.

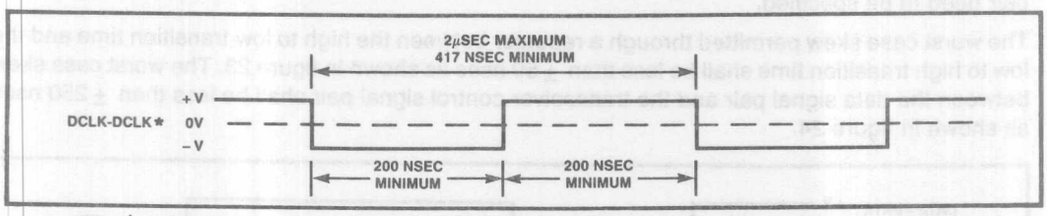


Figure 20. Data Clock Signal Pair Specification for Synchronous Mode

The data signal pair timing is shown in figure 21. The data signal pair is specified with respect to clock edges on the data clock signal pair. Data is changed on the "falling edge" and sampled on the "rising edge" of the data clock pair. These specifications assume balanced delays throughout the system. Specifically, the transmitters for the two signal pairs shall be in the same piece of silicon, the receivers for the two signal pairs shall be in the same piece of silicon and the conductors for the two signal pairs shall be of the same type, the same length and equally loaded.

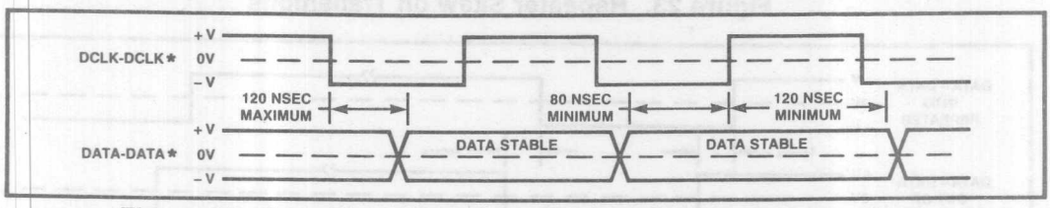


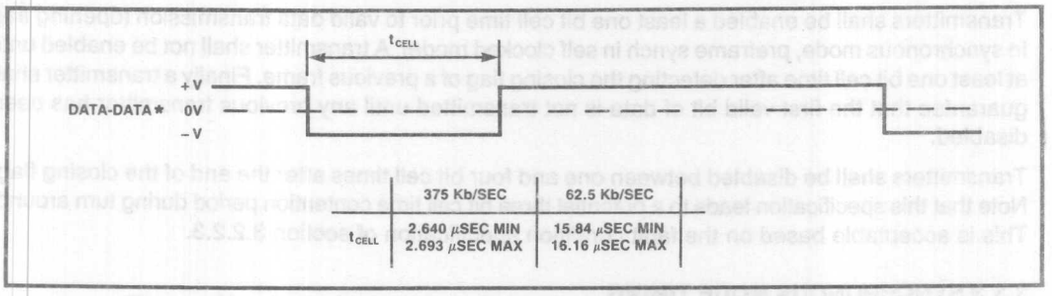
Figure 21. Data Signal Pair Specification for Synchronous Mode

3.3.4 SELF CLOCKED MODE TIMING

The self clocked mode of operation requires specification of the data signal pair and the transceiver control signal pair.

The data signal pair combines both data and clock information in the self clocked mode of operation. The clock information on the signal pair is the transmitter's clock. The receiver uses a separate reference clock to recover the data. The tolerance of each of these clocks shall be ± 1% for both standard bit rates (375 kbit/sec and 62.5 kbits/sec). The transmitter timing is shown in figure 22.

The transceiver control signal pair shall be driven whenever the transmitter is driven. These specifications are provided in section 3.3.2



	375 Kb/SEC	62.5 Kb/SEC
t_{CELL}	2.640 μSEC MIN 2.693 μSEC MAX	15.84 μSEC MIN 16.16 μSEC MAX

Figure 22. Data Signal Pair Specifcation for Self Clocked Mode

3.3.5 REPEATER TIMING

The self clocked mode of operation requires timing specifications for repeaters. Specifically, the skew between the two logic transitions and the skew between the data signal pair and the transceiver control pair need to be specified.

The worst case skew permitted through a repeater between the high to low transition time and the low to high transition time shall be less than ±50 nsec as shown in figure 23. The worst case skew between the data signal pair and the transceiver control signal pair shall be less than ±250 nsec as shown in figure 24.

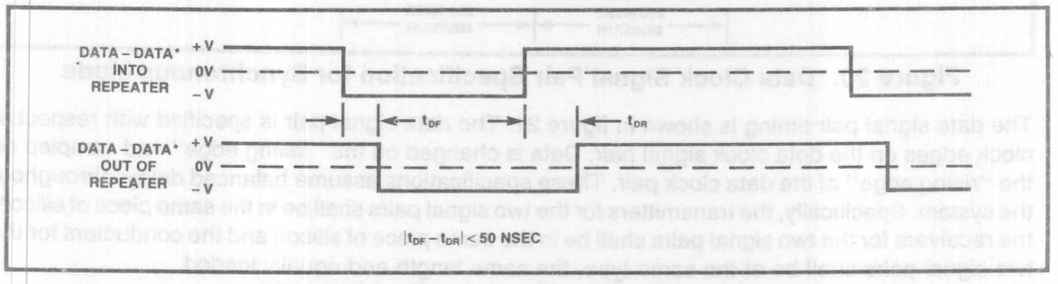


Figure 23. Repeater Skew on Transitions

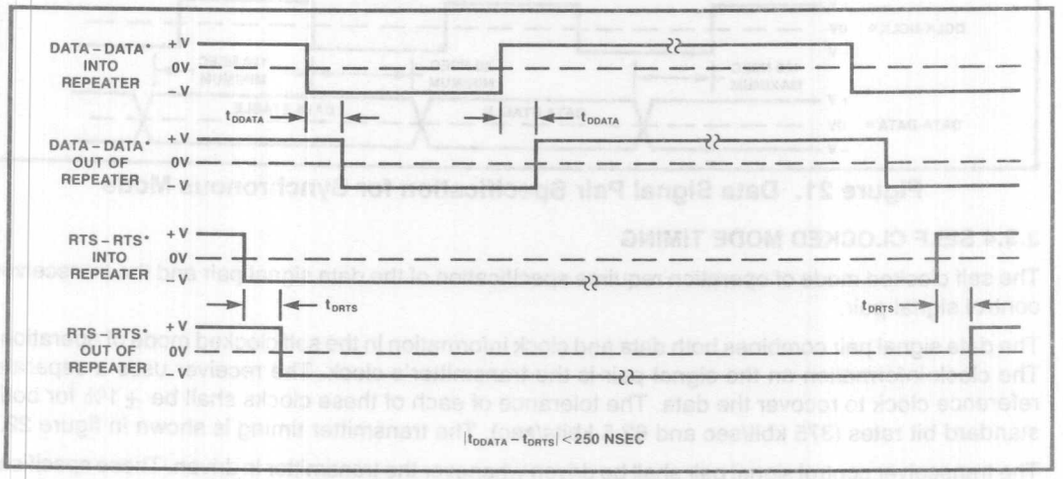


Figure 24. Repeater Skew Between Signal Pairs

The above specifications are for a standard repeater skew. Actual implementations of repeaters may be multiples of or a fraction of this standard. In all cases the specifications shown in table 1 shall apply for the maximum number of standard repeater skews between the master and any slave.

Table 1. NUMBER OF REPEATERS

SPEED (kbit/sec)	MAX # OF STANDARD REPEATERS
375	2
62.5	10

4.0 DATA LINK PROTOCOL SPECIFICATION

The BITBUS data link protocol is a subset of the IBM Synchronous Data Link Control (SDLC) standard. SDLC is used as a basis for the BITBUS data link protocol because it is a highly reliable and proven protocol. The BITBUS interconnect does not (and is not intended to) maintain strict SDLC compatibility.

Based on SDLC, the BITBUS data link protocol connects a single master device to multiple slave devices in a multidrop (i.e. bus) topology. The data link protocol is specifically responsible for encapsulation of messages (messages are defined in section 5 of this specification) into frames and the control of frame transfer over the data link. To accomplish this, the specification defines the concept of system state, a general frame format, specific control fields and basic bus operations.

4.1 SYSTEM STATE

The ability of the BITBUS interconnect to connect nodes over a physically distributed domain requires some aspects of the system state to be formally defined. Being a hierarchical system, this state is defined from the point of view of the master device. This section provides an overview of the concept of system state. A detailed definition is provided in section 4.4.1 of this specification.

4.1.1 MASTER DEVICE STATE

The master device has full knowledge and control of its own state. Operation of a slave device does not require any information about the master device state. This implies that the master device state information is not transferred on the BITBUS interconnect and, therefore, is not a part of this specification.

4.1.2 SLAVE DEVICE STATE

The state of a slave device on the BITBUS interconnect needs to be known by the master device for proper operation. However, being physically separated, it is impossible for the slave device state to be precisely known at all times by the master device (e.g. in the event of a local reset, power down, etc.). In order to update the master device on a slave device state, it is necessary to define a state transfer mechanism over the BITBUS interconnect. The master device uses this mechanism to keep a record of the last known state for each slave device. This state is assumed for the next transfer and checked. If incorrect, the appropriate recovery action is taken. Slave device state consists of two parts: slave device mode and sequence count.

4.1.2.1 Slave Device Mode

A slave device is always in one of two modes: normal disconnect mode (NDM) or normal response mode (NRM). Below is a brief description of each mode. A formal state diagram is provided in section 4.4.1 of this specification.

A slave device enters NDM after a local reset or when it detects an irrecoverable protocol error. In this mode, a slave is awaiting a specific command from the master device to enter NRM. A slave device may not exchange messages with the master device in this mode.

A slave device enters NRM only after receiving a specific command from the master device. Upon entering NRM, a slave device is "synchronized" with the master device, meaning that all sequence counts match (they are all initialized to 0). In this mode, a slave device may exchange messages with the master device as long as "synchronization" is maintained (i.e. no sequence count errors).

4.1.2.2 Sequence Counts

In NRM, sequence counts are used by the master device and each slave device to guarantee that frames are not lost or duplicated. Below is a brief description of how sequencing works. Further details are provided in section 4.3 of this specification.

Sequencing is performed by the master device and each slave device via two pairs of 3 bit sequence counts. Each slave device keeps an N_r (number received) sequence count and an N_s (number sent) sequence count. The master device keeps a corresponding pair of counts for each slave it communicates with. A slave device is "synchronized" with the master device when the sequence counts are correct. The sequence counts at a slave device are considered part of the slave device state. The sequence counts at the master device are master device's best knowledge of slave device state.

The N_r sequence count indicates the sequence count of the next expected incoming message. The N_s sequence count indicates the sequence count of the next message awaiting acknowledgement (may or may not be outstanding). Each time a transfer occurs in NRM these numbers are included and verified resulting in three possible outcomes: correct sequence count, recoverable sequence error or irrecoverable sequence error. The irrecoverable sequence count error case requires the master device to resynchronize with the slave device by causing it to return to NDM then re-enter NRM. The other cases allow the slave device to remain in NRM. Further details are provided later in this specification.

4.2 FRAME FORMAT

The data link protocol is responsible for creating frames to be transferred across the BITBUS interconnect. All frames use the frame format shown in figure 25. All fields are composed of one or more bytes (plus possible zero bit insertion), with the least significant bit (LSB) of each byte transmitted first.

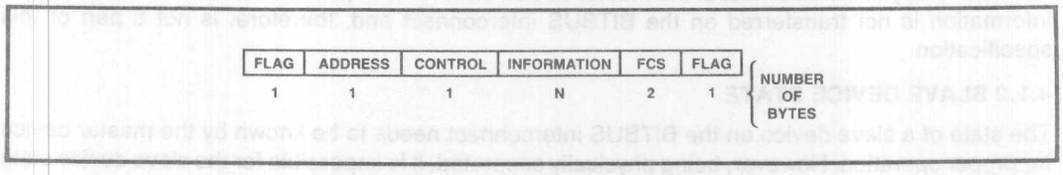


Figure 25. Standard Frame Format

4.2.1 FLAG FIELDS

The flag fields are used to delimit the frame. These fields (one at the beginning of the frame and one at the end of the frame) contain the unique bit pattern 01111110. The uniqueness of this bit pattern is guaranteed by using zero bit insertion/deletion in all other fields as described in section 3 of this specification. These fields are required in all frames.

4.2.2 ADDRESS FIELD

The address field contains the address of the slave device involved in the transfer. For a transmitting master device, this field identifies the destination slave device. For a transmitting slave device, this

field identifies the source slave device to the master device. If this field does not match the slave device address the frame is ignored. The address field is eight bits long and may contain values from 0 to 255. Values 0 and 251-255 are reserved by Intel. All others may be used without restriction. This field is required in all frames.

4.2.3 CONTROL FIELD

The control field is used for command and status exchange between the master device and slave devices. This field is eight bits long and is used for three classes of operations: synchronization, supervision and message transfer. Below is an overview of these operations. Details are provided in section 4.3 of this specification. This field is required in all frames.

4.2.3.1 Synchronization

The transfer of sequenced messages between the master device and a slave device requires that the slave device be properly synchronized to the master device. This synchronization process is performed using unnumbered frames (i.e. frames with unnumbered control fields). As the name implies, unnumbered frames do not use the sequencing feature.

4.2.3.2 Supervision

After the master device is synchronized with a slave device, it is often necessary to exchange status information in the absence of messages. This is done with supervisory frames (i.e. frames with supervisory control fields). These frames are used by the master device to poll a slave device and by a slave device to acknowledge receipt of a valid frame (i.e. address match and no CRC error) from the master device.

4.2.3.3 Information

Information frames (i.e. with information control fields) are used by the master device or a slave device to transfer messages (messages are defined in section 5 of this specification). These frames are only used after synchronization as are supervisory frames. In addition to a message and its sequence count, information frames carry the same status information as supervisory frames. If fact, information frames may be considered as a superset of supervisory frames.

4.2.4 INFORMATION FIELD

The information field is used to carry the BITBUS message. Details of this field are provided in section 5 of this specification. This field is required for information frames and is not used for supervisory or unnumbered frames.

4.2.5 FRAME CHECK SEQUENCE FIELD

The frame check sequence (FCS) field provides the lowest level of error detection on the BITBUS interconnect. This field contains a 16 bit cyclic redundancy check (CRC). The transmitting node generates and sends this field, while the receiving node checks it for correctness. A receiving node ignores an incoming frame if the CRC is incorrect. The CRC is generated by the standard CRC-CCITT polynomial $X^{16} + X^{12} + X^5 + 1$. This field is required in all frames.

4.3 CONTROL FIELD DEFINITION

Operations on the BITBUS interconnect are performed using three types of control fields: unnumbered, supervisory and information. This section specifies these fields in detail and explains their usage.

For reference to SDLC, this specification uses a subset of the defined control fields with the poll/final bit always set. This implies that frames sent by the master device always expect a response from

the addressed slave device (poll bit set) and that frames sent by slave devices always return control of the link (final bit set) to the master device.

4.3.1 UNNUMBERED FRAMES

Unnumbered frames are used on the BITBUS interconnect for synchronizing slave devices with the master device. This section specifies the control field format for these frames and defines the unnumbered operations (commands and responses) supported on the BITBUS interconnect.

4.3.1.1 Unnumbered Control Field Format

The control field format for unnumbered frames is shown in figure 26. This field may specify one of several unnumbered operations.

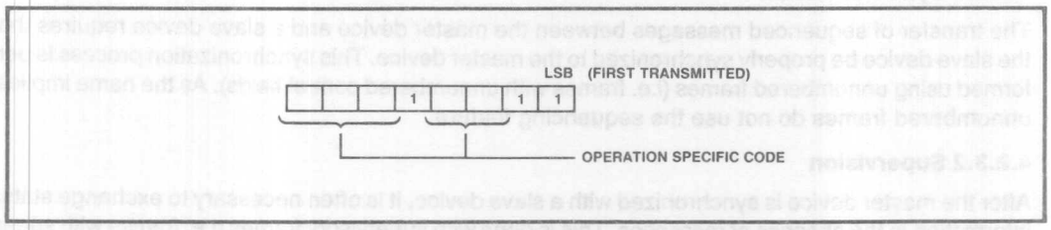


Figure 26. Unnumbered Control Field Format

4.2.1.2 Unnumbered Operations

The BITBUS interconnect supports two unnumbered commands (SNRM and DISC) and two unnumbered responses (UA and FRMR). The control fields for these operations are summarized in table 2.

Table 2. UNNUMBERED CONTROL FIELDS

OPERATION	COMMAND	RESPONSE	CONTROL VALUE FIELD
SNRM	✓		93H
DISC	✓		53H
UA		✓	73H
FRMR		✓	97H

The frame reject (FRMR) command is sent to the master device by a slave device that detects an invalid control field in an otherwise valid frame. The response is used to respond to any unnumbered frame while in NRM, any supervisory or information frame while in NDM, any unsupported control field, or an irrecoverable sequence count error. Upon receiving this command, the master device shall initiate resynchronization with the slave device.

The unnumbered acknowledge (UA) response is used by a slave device to acknowledge receipt of a valid unnumbered command while in NDM.

The disconnect (DISC) command is sent by the master device to a slave device to initiate resynchronization. This command causes the slave device to go to, or stay in, NDM. It is used by the master device when it detects the need to resynchronize (e.g. after reset, irrecoverable sequence error, etc.) or in response to an FRMR from a slave device. When the master device receives an UA in response to a DISC it knows that the addressed slave device is in NDM.

The set normal response mode (SNRM) command is sent by the master device to synchronize a slave device. If a slave device is in NDM, this command causes it to enter NRM, allowing it to exchange messages with the master device. If a slave device is already in NRM, this command is invalid.

4.3.2 SUPERVISORY FRAMES

Supervisory frames are used on the BITBUS interconnect for passing status between the master device and a slave device while in NRM. These frames are used by a master device to poll slave devices and by slave devices to acknowledge receipt of a valid frame from the master device. This section specifies the control field format for these frames and defines the specific control fields that are supported.

4.3.2.1 Supervisory Control Field Format

The control field format for supervisory frames is shown in figure 27. This field is used to specify the supervisory operation (RR or RNR) as well as to acknowledge receipt of a previous message via the N_r sequence count.

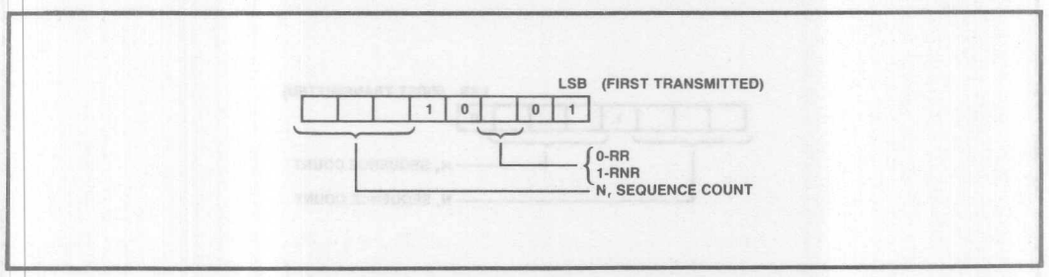


Figure 27. Supervisory Control Field Format

4.3.2.2 Supervisory Operation

The BITBUS interconnect supports two supervisory operations: receiver ready (RR) and receiver not ready (RNR).

The master device uses the RR supervisory frame to poll a slave device for an information frame. Slave devices use the RR supervisory frame to acknowledge valid reception (i.e. address match, valid CRC and available buffer space) of a previous supervisory or information frame when a message is not available (If a message is available an information frame is sent unless the incoming frame was a RNR supervisory frame).

The RNR supervisory frame is used to indicate that a buffer is presently unavailable to receive an otherwise valid frame. The master device uses the RNR supervisory frame to poll a slave device for data link status only. A slave device shall not respond to a RNR supervisory frame with an information frame. A slave device uses the RNR supervisory frame to indicate that the last frame was recognized, however there was no buffer available to store it. In this case the master device shall retransmit the frame.

Both RR and RNR frames contain an N_r sequence count. This sequence count acknowledges receipt of N_r-1 frames (i.e. N_r equals the next expected incoming frame). The receiver of a supervisory frame shall always compare this value to its N_s sequence count for correctness.

In the case that all messages have been acknowledged, the incoming N_r sequence count should equal the N_s sequence count. If the previous frame sent by the receiving node was an information frame, then the incoming N_r sequence count should equal the N_s sequence count plus 1. If this case is true, the previous frame is acknowledged meaning the N_s sequence can be incremented and the message buffer released. If after an information frame the incoming N_r sequence count is equal to the N_s sequence count, the information frame was not received and shall be retransmitted. Any other N_r sequence count value is detected as an irrecoverable sequence error requiring the master device to resynchronize with the slave device.

4.3.3 INFORMATION FRAMES

Information frames are used on the BITBUS interconnect for message transfer. These are the only frames that contain an information (I) field. The content of the I field is the BITBUS message as defined in section 5 of this specification. This section specifies the control field format for these frames and defines the operation of the sequence counts.

4.3.3.1 Information Control Field Format

The control field format for information frames is shown in figure 28. The control field contains an N_r sequence count to acknowledge valid reception of $N_r - 1$ frames and an N_s sequence count corresponding to the attached information field (i.e. message).

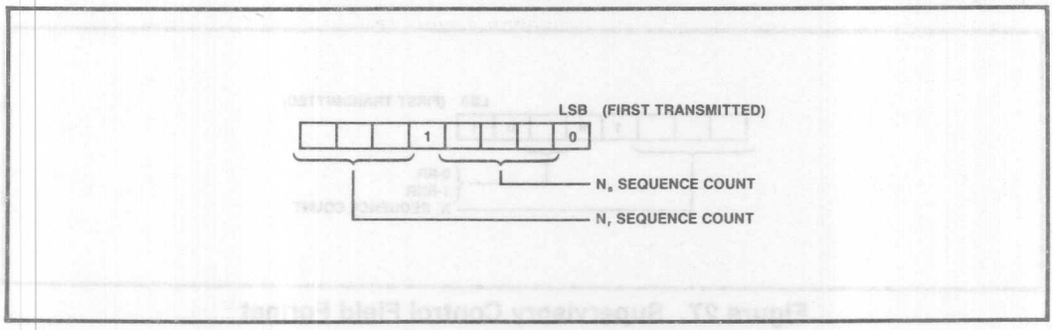


Figure 28. Information Control Field Format

4.3.3.2 Information Operation

Information frames are a superset of supervisory frames. They carry the same acknowledgement status as supervisory frames via the N_r sequence count. This count shall always be compared to the receiver's N_s sequence count as it is for supervisory frames.

In addition to the N_r sequence count, information frames carry a message in the information field and a corresponding N_s sequence count. Upon receiving an information frame, a receiver shall compare the incoming N_s count with its N_r count. If the counts match, the message shall be passed on. If the counts do not match, error recovery action shall proceed as follows. If the incoming sequence count N_s equals the receiver's sequence count $N_r - 1$, the receiving node may acknowledge receipt of the frame, then discards it (this case occurs when an information frame is received and its acknowledge is corrupted, making the second information frame a duplicate). If the incoming sequence count N_s does not equal the receiver's sequence count N_r or $N_r - 1$, then an irrecoverable sequence error is detected. This case requires the master device to resynchronize with the slave device before further message transfers can occur.

4.4 DATA LINK OPERATION

This section summarizes the data link operation on the BITBUS interconnect. This summary includes a precise definition of slave state, slave state transition, and sequenced operations in NRM. In addition sample synchronization sequences, sample transfer sequences and a summary of error conditions and the corresponding recovery actions are presented.

4.4.1 SLAVE STATE DESCRIPTION

The state transition diagram for a slave device is shown in figure 29. As previously discussed, the state diagram has only two basic states: normal response mode (NRM) and normal disconnect mode (NDM). Table 3 lists the responses to all possible incoming frames in each of the states. Note that

this table assumes that the slave device has recognized a valid incoming frame. That is, the address field matched the node address and the CRC field was correct.

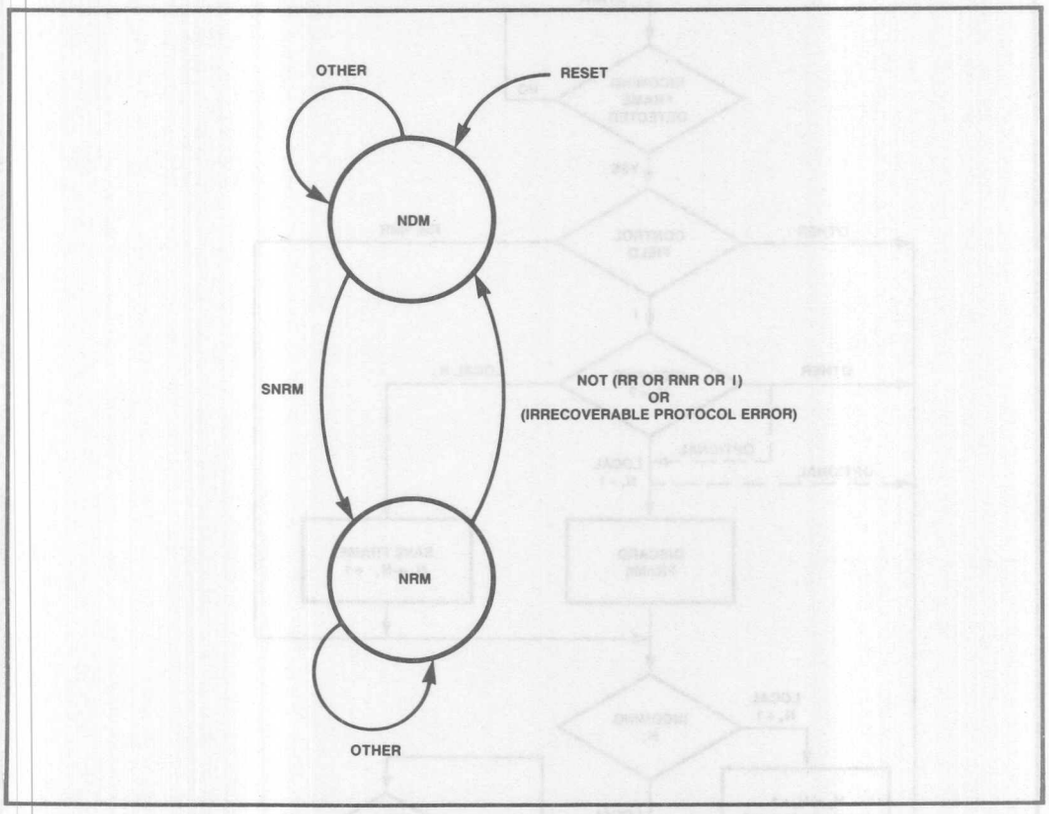


Figure 29. Slave Device State Diagram

Table 3. SLAVE DEVICE STATE TRANSITIONS AND RESPONSES

SLAVE STATE	INCOMING FRAME	NEXT STATE	RESPONSE
NDM	DISC	NDM	UA
NDM	SNRM	NRM	UA
NDM	Other	NDM	FRMR
NRM	INFORMATION	NRM	RR,RNR,I
NRM	RR	NRM	RR,RNR,I
NRM	RNR	NRM	RR,RNR
NRM	INFORMATION,RR,RNR with irrocoverable sequence error	NDM	FRMR
NRM	Other	NDM	FRMR

When in NRM, the slave device state also includes its N_r and N_s sequence counts. For simplicity, these states are not shown but instead, a flow chart is provided in figure 30. This flow chart, along with table 3, completely specifies the slave device interface.

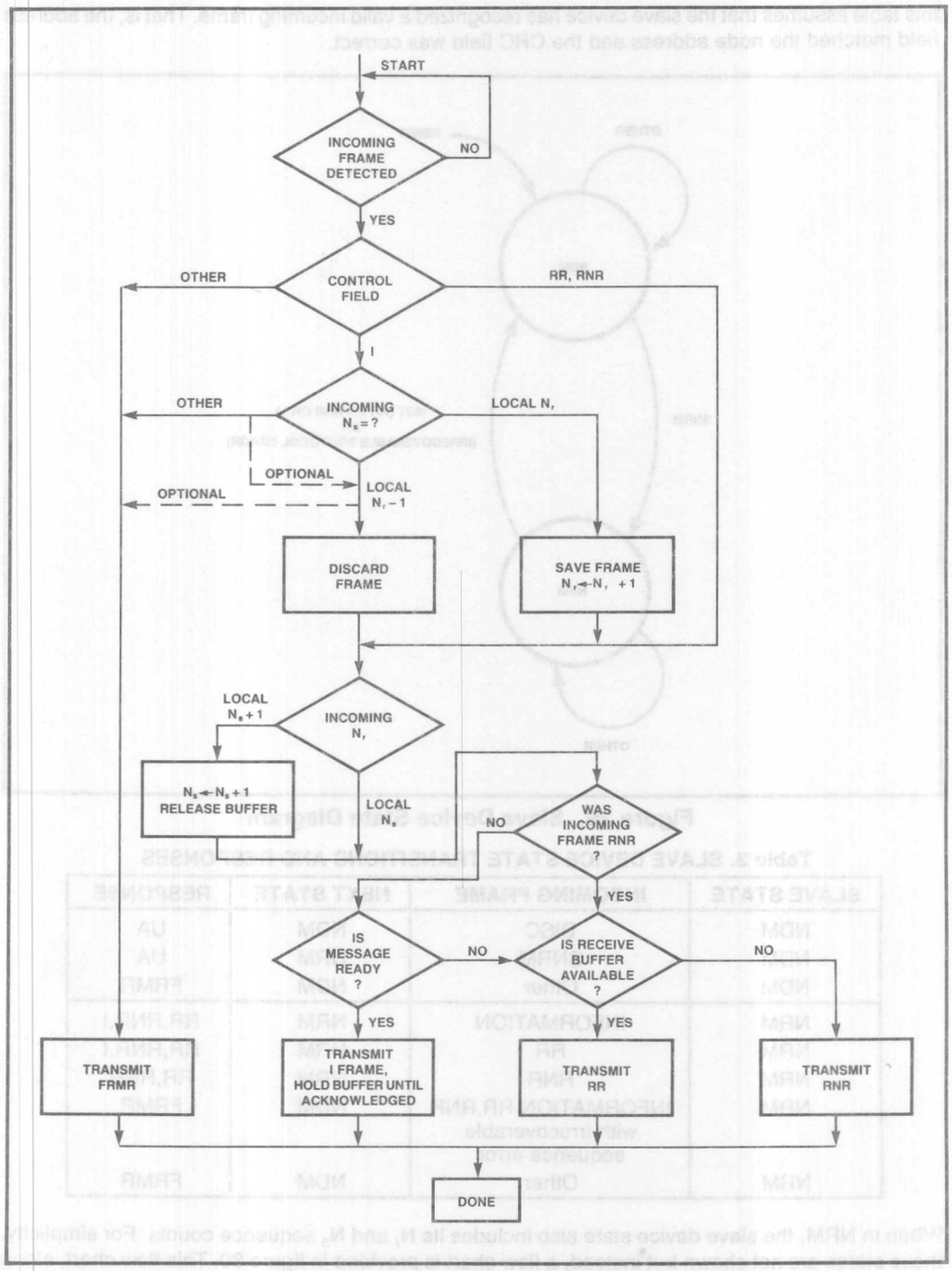


Figure 30. Slave Response to Incoming Frame in NRM

4.4.2 SYNCHRONIZATION SEQUENCE

The synchronization sequence on the BITBUS interconnect is used to establish a known state at the slave device (NRM with sequence counts equal to zero), in order to allow sequenced message transfer. Synchronization is required anytime the master device is reset or anytime the master device or slave device detect an irrecoverable protocol error. Synchronization is strictly a data link protocol feature and, therefore, does not appear at any higher levels of this specification. Figure 31 shows a typical synchronization sequence.

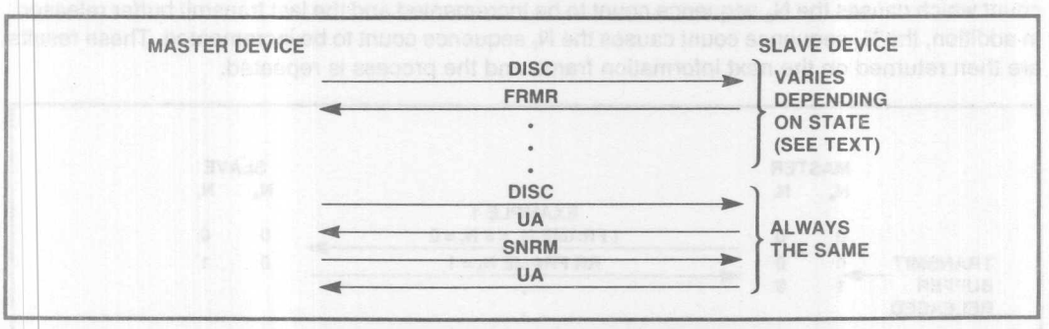


Figure 31. Typical Initialization Sequence

The master device may initiate the synchronization sequence by sending a DISC. The response from the slave device to a disconnect is a FRMR if it is in NRM or an UA if it is in NDM. The master device shall continue sending DISC frames until a UA response is returned verifying that the slave is in NDM. The master device then completes the sequence by sending an SNRM. Upon receiving the SNRM while in NDM, a slave device enters NRM and responds with a UA. Once the master device receives a UA to an SNRM, it knows that the slave device is synchronized.

The slave device may initiate the synchronization sequence by responding to an incoming frame with a FRMR frame. The master device responds to a FRMR with a DISC and proceeds through the sequence described above.

4.4.3 TRANSFER SEQUENCES

This section provides several transfer sequence examples. These examples serve as a basis for the message protocol defined in section 5 of this specification. Three examples are provided in figure 32: message transfer from a master device to a slave device, message transfer from a slave device to a master device and piggybacking of messages on acknowledges or polls. For these examples, it is assumed that the slave device is synchronized and that incoming frames are valid. Errors are discussed further in section 4.4.4 of this specification.

In the first example, the master device sends a message to a slave device. The information frame that carries the message also carries the sequence counts $N_s = 0$ and $N_r = 0$. The slave device verifies that the incoming N_r equals its N_s and that the incoming N_s equals its N_r. The slave then increments N_r to indicate it received the information frame and returns an RR frame with sequence count $N_r = 1$ to acknowledge the frame. Upon receiving the N_r sequence count, the master device sees that its information frame has been received, increments its N_s sequence count for the next message and releases the transmit buffer.

In the second example, the master device is polling a slave device with an RR frame for a message. Upon receiving the RR frame, the slave device verifies the incoming N_r sequence count and responds with an information frame containing the message and sequence counts $N_s = 0$ and $N_r = 1$.

Upon receiving the information frame, the master device verifies the incoming N_r and N_s sequence counts. It then increments its N_r to indicate that it has received the information frame. On the next transmission to the slave device, the master device sends its sequence count $N_r = 1$ to acknowledge receipt of the information frame. Upon detecting this, the slave device increments its N_s sequence count and releases the transmit buffer.

In the third example, the link efficiency is increased by piggybacking acknowledges and polls onto information frames. At both master device and slave device, an incoming frame carries an N_r sequence count which causes the N_s sequence count to be incremented and the last transmit buffer released. In addition, the N_s sequence count causes the N_r sequence count to be incremented. These results are then returned on the next information frame and the process is repeated.

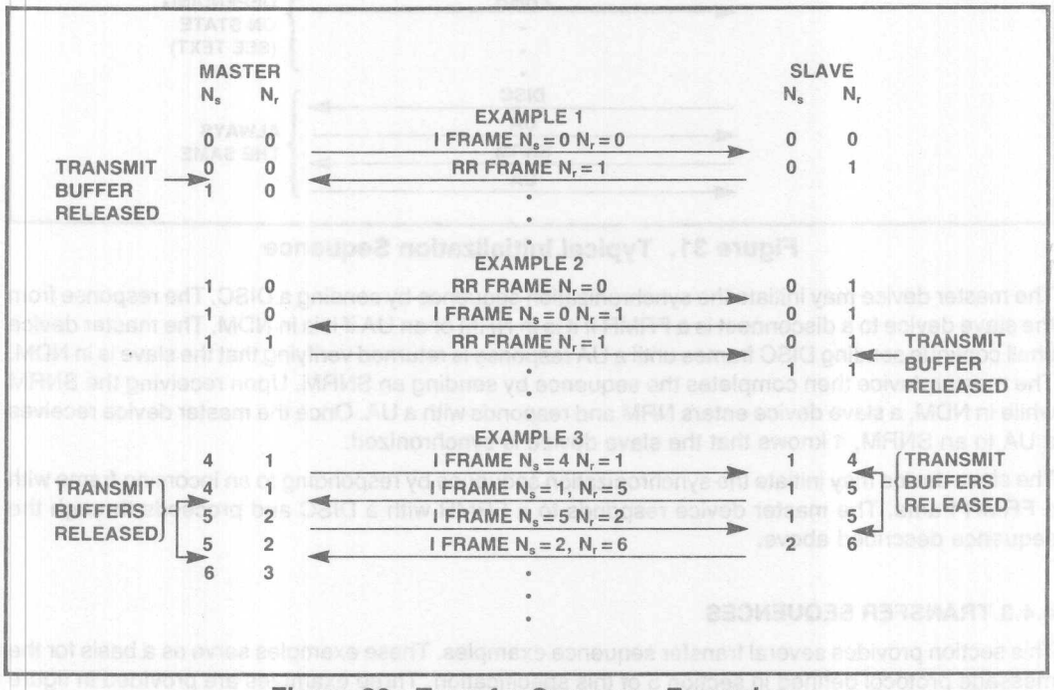


Figure 32. Transfer Sequence Examples

4.4.4 DATA LINK ERROR SUMMARY

The following section summarizes the possible data link error conditions and specifies the corresponding action. Specific error conditions are time out, invalid control field and sequence count.

4.4.4.1 Time Out

A time out error occurs when the master device does not receive a response to a transmitted frame within 10 milliseconds. This time is measured from the time transmission is completed until the response is received and the CRC verified.

Time out errors can occur under the following conditions: the addressed slave does not exist, the address slave discards the frame due to a CRC error, or a CRC error is detected by the master device in a response. In all cases the recovery action taken by the master device is to retransmit the frame a second time. If two consecutive errors are detected an irrecoverable protocol error is returned to the user.

4.4.4.2 Invalid Control Field

An invalid control field detected by either the master device or a slave device is considered an irrecoverable protocol error. Detection of this error condition at a slave device causes the slave device to enter NDM and respond with a FRMR. Detection at the master device causes it to assume that the slave is in NDM, requiring resynchronization.

4.4.4.3 Sequence Count

Sequence count errors are recoverable in some cases as indicated in figure 30. Cases that are not recoverable are considered irrecoverable protocol errors. Detection of an irrecoverable condition at a slave device causes it to enter NDM and respond with a FRMR. Detection at the master device causes it to assume that the slave is in NDM, requiring resynchronization.

5.0 MESSAGE PROTOCOL

The BITBUS message protocol is designed to provide a task to task message interface between a master node and multiple slave nodes using an order/reply structure. That is, the master node issues orders to slave nodes which respond with replies. This structure is built on top of the data link protocol using information frames for the message transfer. This section specifies the order/reply structure (including error handling) and the standard message format.

5.1 ORDER/REPLY STRUCTURE

The BITBUS message protocol is based on an order/reply structure in a multitasking environment. An order is defined as a message that is sent by a task on the master node, over the BITBUS interconnect, to a task on a slave node. A reply is defined as a message that is sent by as task on a slave node, over the BITBUS interconnect, to a task on the master node, in response to an order from that master node task. Master node tasks may exist on either the master device or master device extension (collectively referred to as the master node). Likewise, slave node tasks may exist on either the slave device or slave device extension (collectively referred to as the slave node). Note that orders and replies may have a more general definition when used between tasks on a given node. This special case is not part of this specification.

Every order on the BITBUS interconnect requires a corresponding reply; however, replies need not be returned in the same sequence that orders are issued. For example, if the master node issues orders A,B,C to a slave node, the responses may be returned as A,B,C or B,C,A or C,A,B or etc. Based on this requirement, a master device need only poll a slave node when one or more orders are outstanding. This condition exists when the sequence counts N_r and N_s for a given slave node are not equal. When the sequence counts N_r and N_s are equal, the slave node has no more replies and need not be polled. This algorithm limits unnecessary polling and increases system performance.

The data link protocol provides sequence counts that are 3 bits long. This limits the number of outstanding messages to any given slave node to seven. The slave node is responsible for enforcing this limit by not providing a receive buffer for the eighth message.

The BITBUS message protocol provides sufficient control to route orders from the master node to a slave device or slave device extension and replies from a slave node to a master device or master device extension. This capability allows local messages within a node to co-exist in a common gateway implementation. Furthermore, multiple level hierarchies (as shown in figure 1) are easily supported using the same master device software. The details of these implementations are not part of this specification.

Several sources of error exist in the message protocol. Specific examples are irrecoverable protocol errors or missing tasks. In all cases, the BITBUS message protocol detects the errors and handles

them in the same way. If an error occurs while delivering an order, the complete order is immediately converted to a reply and returned to the originating task. The reply is identical to the order except for the error code in the response field. If an error is detected on a reply, it is simply discarded (note that this is a very rare case, except for catastrophic error, since the reply retraces the path of the order). In either case, there is a small possibility that a reply is not returned. The task originating the order must account for this by setting a time out period for recovery. Note that this time out is not related to the data link time out specified in section 4 of this specification. Its value is determined by the application.

5.2 MESSAGE FORMAT

The standard message format on the BITBUS interconnect is shown in figure 33. This message structure is sent across the BITBUS interconnect in the information field of information frames. All fields shown, except the data field, are required in all messages.

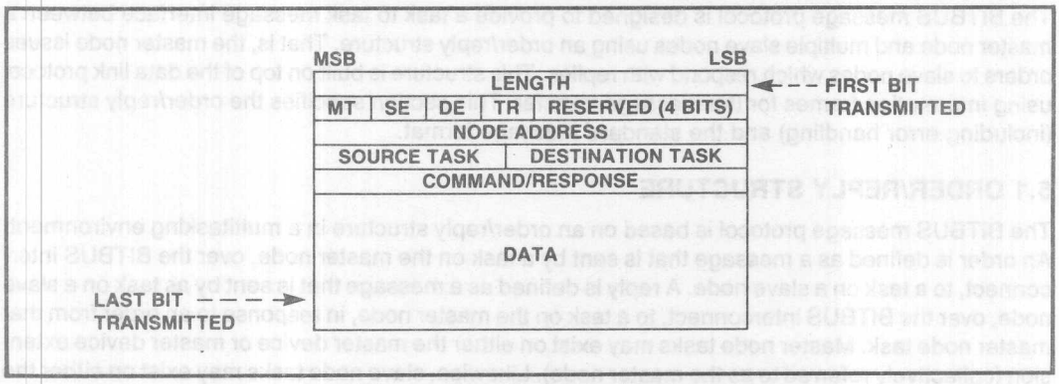


Figure 33. Message Format

5.2.1 LENGTH

The length field specifies the total message length. This eight bit field may contain values between 7 and 255. The value in this field equals the number of bytes in the data field plus 7. This value allows implementation to easily add two bytes for local message manipulation such as buffer control or queuing. All implementations shall support a data field of up to 13 bytes, corresponding to a length field equal to 20.

5.2.2 MESSAGE TYPE (MT)

The message type field is used to specify whether the message is an order or a reply. The master node always sends orders and, therefore, shall set this bit to 0. A slave node always sends replies and, therefore, shall clear this bit to 1.

5.2.3 SOURCE EXTENSION (SE)

The source extension field indicates whether the source of an order, or the destination of a reply, is the master device or the master device extension. This bit is set to 1 to indicate the master device extension and cleared to 0 to indicate the master device. This bit is unchanged between an order and its corresponding reply.

5.2.4 DESTINATION EXTENSION (DE)

The destination extension field indicates whether the destination of an order, or source of a reply, is a slave device or a slave device extension. This bit is set to 1 to indicate a slave device extension

and cleared to 0 to indicate a slave device. This bit is unchanged between an order and its corresponding reply.

5.2.5 TRACK (TR)

The track field is used to provide message control at a master or slave device which may be required by some implementations. This bit shall be cleared to 0 when sending a message. It shall be set to one upon receiving a message from the BITBUS interconnect.

5.2.6 RESERVED

These four bits are reserved for possible future enhancements. They shall be cleared to zero when sending a message and their value is not guaranteed upon receiving a message.

5.2.7 NODE ADDRESS

The node address field specifies the destination node for orders and the source node for replies (i.e. it is unchanged between order and reply). This eight bit field contains the same value as the address field in the data link frame. Valid entries are 1 through 250. The values 0 and 251-255 are reserved by Intel.

5.2.8 SOURCE TASK

The source task field identifies the task that has generated an order or is to receive a reply. This four bit field allows up to sixteen tasks to generate orders from the master device and the master device extension. Specific implementations may support as few as one or as many as sixteen tasks.

5.2.9 DESTINATION TASK

The destination task field identifies the task that is to receive an order or has generated a reply. This four bit field allows up to sixteen tasks to receive orders at each slave device and slave device extension. Specific implementations may support as few as one or as many as sixteen tasks.

5.2.10 COMMAND/RESPONSE

The command/response field is used by both user tasks and the message protocol. Under normal conditions, this field is used only by the user tasks. The message protocol only uses this field for reporting errors. Table 4 indicates the usage of this field. Note that this table only applies to responses. Commands are always defined by the user task.

Table 4. MESSAGE PROTOCOL RESPONSES

CONDITION	ERROR
No error (set by user)	00H
User defined	01H-7FH
No destination task	80H
Protocol Error	91H
No destination device	93H
Reserved by Intel	81H-90H,92H, 94H-0FFH

5.2.11 DATA

The data field is defined by the contents of the command field. Minimum support requires this field to be capable of handling 13 bytes. Implementation may extend it to as much as 248 bytes provided that the longer messages are not sent to nodes that cannot support them. This field is the only optional field in the message.

6.0 REMOTE ACCESS AND CONTROL

The remote access and control (RAC) interface for the BITBUS interconnect defines a set of high level commands and responses to perform general purpose operations at a slave node. The remote access interface allows I/O read and write capabilities as well as memory download and upload. The remote control interface allows control and monitoring of tasks at a slave node. The RAC interface is built on top of the message protocol defined in section 5 of this specification. In fact, RAC is simply a special task defined at slave nodes. This section specifies the commands, responses and the general RAC message format.

6.1 MESSAGE FORMAT FOR RAC

The RAC interface is built on top of the standard message protocol. The message format follows the standard format in figure 33 with the following fields further defined: destination task, command/response, and data.

6.1.1 DESTINATION TASK

The BITBUS interconnect, by convention, defines task 0 as the RAC task. This makes its presence easily detectable and allows it to be addressed in the same way for all implementations. If the RAC task is not supported at a slave node, task 0 may be used for another function.

6.1.2 COMMAND/RESPONSE

The command/response field is used to identify the RAC command in an order message and the RAC response in reply messages. The various commands and responses supported by the RAC interface are specified in sections 6.2 and 6.3, respectively.

6.1.3 DATA

The contents of the data field are determined by the RAC command. The length field is used to identify the exact number of data bytes. The minimum BITBUS interconnect support provides up to 13 bytes in this field. Details of this field are provided in sections 6.2 and 6.3 of this specification.

6.2 RAC COMMANDS

Table 5 lists the RAC commands supported on the BITBUS interconnect. Commands 00H through 0BFH are general purpose commands controlled by Intel. At this time, 15 are specifically defined, and the rest are reserved. Commands 0C0H through 0FFH are available to the user for custom RAC commands. These values will not be standardized. Specific implementations of RAC may support all or any subset of the defined commands. The only specific requirements are that the implementation return an error response for unsupported commands and that all supported general purpose commands conform to this specification.

6.2.1 CREATE TASK

The create task command is used to initialize and begin execution of a task at a slave node. This command assumes that the task to be created is already in the slave node memory. The command is simply the mechanism used to inform the operating system at the slave node that a given task in memory is to become active.

The data field in a create task order contains an address pointer (length field equal to 7 plus pointer length) to the task descriptor in the slave node's memory. The length of the pointer is determined by the implementation requirements at the slave node. This pointer is sent most significant byte first. The actual task descriptor may vary between implementations and, therefore, is not part of this specification.

Table 5. RAC COMMANDS

COMMAND	ACCESS	CONTROL	VALUE
Reset Slave		✓	00H
Create Task		✓	01H
Delete Task		✓	02H
Get Function ID		✓	03H
RAC Protect		✓	04H
Read I/O	✓		05H
Write I/O	✓		06H
Update I/O	✓		07H
Upload Memory	✓		08H
Download Memory	✓		09H
OR I/O	✓		0AH
AND I/O	✓		0BH
XOR I/O	✓		0CH
Status Read	✓		0DH
Status Write	✓		0EH
Reserved			0FH-0BFH
User Defined			0C0H-0FFH

After successful completion of a create task command, a reply message is returned to the originating task on the master node with 00H in the response field and the data field unchanged (i.e. address pointer is returned).

6.2.2 DELETE TASK

The delete task command performs the inverse function to the create task command. It is used to indefinitely terminate execution of a task at a slave node. This allows the task number associated with the deleted task to be reused (i.e. with a create task command).

The data field in a delete task order contains a single byte (length field equal to 8) identifying the task number to be deleted. This field may contain values between 0 and 15; however, beware of deleting task 0 as it is the RAC task.

After successful completion of a delete task command, a reply message is returned to the originating task on the master node with 00H in the response field and the data field unchanged (i.e. eight bit task number is returned).

6.2.3 GET FUNCTION IDs

Function IDs are logical IDs used to identify a given task by its function rather than its physical address. The concept of function ID allows functions to maintain a constant identifier, even if they are moved to a new physical address (node address and/or task number). The get function IDs command causes the slave device, or slave device extension, to respond with a list of function ID codes for the tasks currently in existence.

The get function IDs order message carries a dummy data field equal in length to the number of function IDs to be returned. The length field is used to identify the number of bytes (value in length field equals number of bytes plus 7). Carrying the dummy data field allows the slave node to allocate a single buffer that is sufficiently large to receive the order message, as well as return the reply message.

After execution of the get function IDs command, a reply message is returned to the originating task on the master node with 00H in the response field and the function ID codes in the data field. This first data byte corresponds to task 0, the second data byte to task 1, etc. The number of data bytes returned is always equal to the number sent in the order message.

The assignment of function IDs is summarized in table 6. As an example, consider a slave device that supports up to 8 tasks. Task 0 is RAC, task 1 is a user task with function ID 81H, tasks 2 and 3 are present but have no function ID and tasks 4-7 are not used. For this case, the length field in the order message and reply message is 15. The reply message would have a data field as shown in figure 34.

Table 6. FUNCTION ID CODE ASSIGNMENTS

FUNCTION	VALUE
No Task	00H
RAC Task	01H
Reserved by Intel	02H-7FH
User Assigned	80H-0FEH
Task with no Function ID	0FFH

MSB LSB

LENGTH				
MT	SE	DE	TR	RESERVED (4 BITS)
NODE ADDRESS				
SOURCE TASK			DESTINATION TASK	
COMMAND/RESPONSE				

TASK NUMBER	TASK FUNCTION
0	RAC
1	User Function 81H
2	Task without function ID
3	Task without function ID
4	No Task
5	No Task
6	No Task
7	No Task

Figure 34. GET Function ID Reply Example

6.2.4 RAC PROTECT

The RAC protect command allows the remote access functions at a slave node to be enabled or disabled by a task on the master node. When disabled, the RAC function at a slave node only recognizes the remote control commands. On reset, the RAC function shall be enabled.

The RAC protect order contains a single byte data field and a length field of 8. If the value of this data field is 00H, the RAC function is enabled. If the value in this data field is 01H, the RAC function is disabled. Any other value may produce indeterminate results and should not be used.

After successful completion of an RAC protect command, a reply message is returned to the originating task on the master node with 00H in the response field and the data field unchanged.

6.2.5 RESET SLAVE

The reset slave command initiates a reset at the addressed slave device or slave device extension. The extent of the reset (e.g. software, slave device, slave device extension, entire slave node, etc.) is determined by the implementation. The data field for this command is null and the length field is always 7.

The reset slave command is the only command for which a reply message is not returned upon successful completion. A reply message is not returned in this case since its delivery cannot be predicted while the slave node is in the process of resetting. The only case when a reply message is attempted is in the case of error. This case is discussed in section 6.3 of this specification.

6.2.6 MEMORY COMMANDS

The memory commands allow blocks of data to be moved between the master node and a slave node. Operations include memory upload and memory download. Since these commands share a common format in the message data field, they are presented together.

The general data field format for memory commands is shown in Figure 35. This format includes a 16 bit address pointer and 1 to n bytes of data. The length field in the header is used to specify the actual number of data bytes (e.g. length equal to 9 plus n for n data bytes). The address pointer is used to identify the base address for the data bytes. That is, the pointer is the address associated with data byte 1 and the address of subsequent data bytes is obtained by simply incrementing the pointer.

The address pointer is limited to a 64K address range as this is believed to be sufficient for the vast majority of BITBUS interconnect applications. In the few cases where a larger address range is needed, it can be created by assigning a location in the I/O or status space as an offset register, thus allowing the memory commands to operate in 64K byte segments or pages.

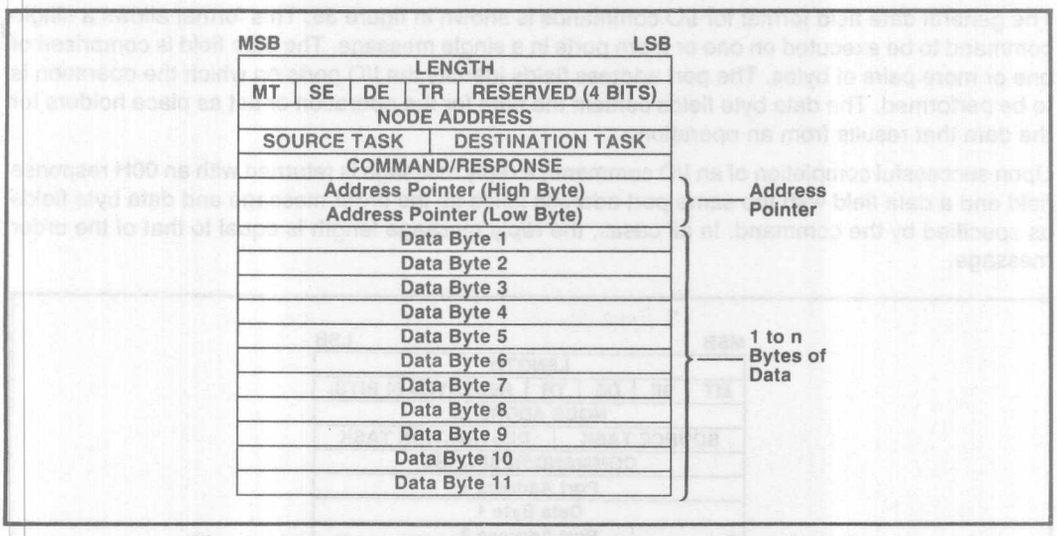


Figure 35. Data Field Format for Memory Commands

6.2.6.1 Memory Upload

The memory upload command causes a slave device, or slave device extension, to read a specified number of data bytes from its local memory and return them in a reply message. The specific number of bytes to be read is identified in the length field of the order message (i.e. length field equals number of bytes to be read plus 9).

The data field in a memory upload order message contains a 16 bit address pointer and n invalid data bytes, where n is the number of bytes requested. The invalid data bytes are carried to simplify buffer allocation at the slave node by allowing a single buffer to be allocated that is sufficient for delivering the order and returning the reply.

At the slave node, the RAC task reads sequentially the specified number of data bytes. The first byte is read from the location specified by the address pointer and subsequent bytes are located by incrementing the address pointer. Upon successful completion, a reply message is returned with an 00H response field and a data field with the original address pointer and the requested data bytes.

6.2.6.2 Memory Download

The memory download command causes a slave device, or slave device extension, to write n bytes of data into its local memory. The specific number of bytes to be written is identified in the length field of the order message (i.e. length field equals number of bytes to be written plus 9). In addition to the data bytes, the data field contains a 16 bit address pointer.

At the slave node, the RAC task writes the data bytes sequentially starting at the location specified by the address pointer. Subsequent addresses are located by incrementing the address pointer for each byte. Upon successful completion, a reply message is returned with an 00H response field and a data field containing the contents of the order message.

6.2.7 I/O COMMANDS

The I/O commands allow the master node to access up to 256 I/O ports on each slave device and each slave device extension. Operations include read I/O, write I/O, update I/O, OR I/O, AND I/O, and XOR I/O. Since these commands all share a common format in the message data field, they are presented together.

The general data field format for I/O commands is shown in figure 36. This format allows a single command to be executed on one or more ports in a single message. The data field is comprised of one or more pairs of bytes. The port address fields identify the I/O ports on which the operation is to be performed. The data byte fields contain the data for the operation or act as place holders for the data that results from an operation.

Upon successful completion of an I/O command, a reply message is returned with an 00H response field and a data field with the same port address fields as the order message and data byte fields as specified by the command. In all cases, the reply message length is equal to that of the order message.

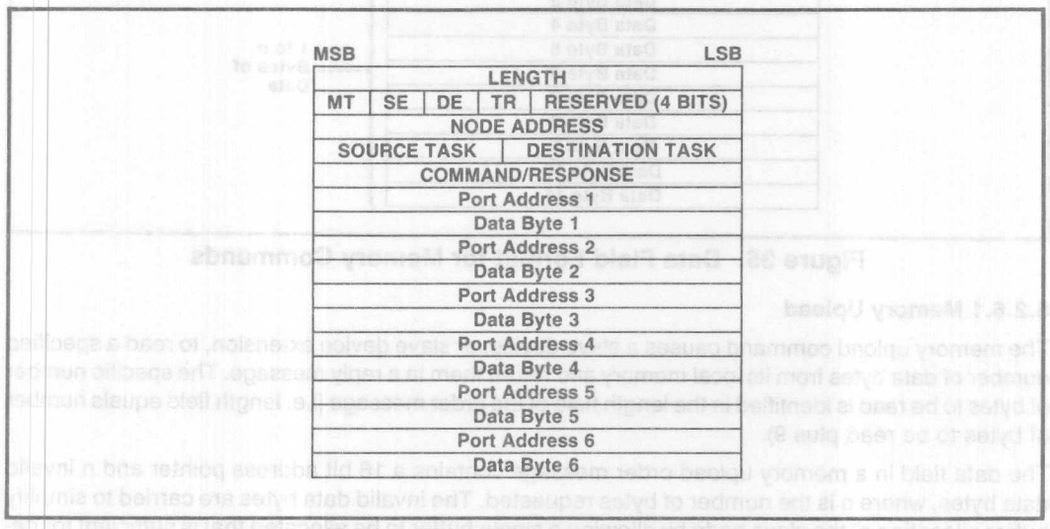


Figure 36. Data Field Format for I/O Commands

6.2.7.1 Read I/O

The read I/O command causes a slave node to read the specified I/O ports. The data byte fields in the order message are undefined. They are carried only to simplify buffer allocation at the slave node. The data byte fields in the reply message contain the data read from the specified ports.

6.2.7.2 Write I/O

The write I/O command causes a slave node to write the data bytes fields to the specified I/O ports. The data byte fields in the order message contain the data to be written. The data byte fields in the reply message are unchanged from the order message.

6.2.7.3 Update I/O

The update I/O command causes a slave node to write the data byte fields to the specified I/O ports then reread the ports. The data byte fields in the order message contain the data to be written. The data byte fields in the reply message contain the data reread from the I/O port after the write operation.

6.2.7.4 Logical Operations

The logical operation commands provide bit set, bit reset and bit toggle operations. The data byte fields for these operations contain a mask. The command causes the slave node to read the specified I/O port, perform the specified logical operation with the mask in the data byte field, write the result back to the I/O port, then finally reread the port. This final value is entered in the data byte field for the reply message.

Bit operations are defined to logically set, clear and toggle I/O bits. The electrical levels of these operations depends on the implementation. When no signal inversion exists between the internal CPU state and the I/O pin, the operations of set and clear correspond to the electrical high and low states respectively. These implementations are referred to as "active high." If a signal inversion occurs between the internal CPU state and the I/O pin, the electrical definitions of set and clear are reversed. These implementations are referred to as "active low."

From the logical point of view, bit operations are defined as follows. A bit is set by using an OR I/O command with a one in the bit position to be set. A bit is cleared by using an AND I/O command with a zero in the bit position to be cleared. A bit is toggled by using an XOR I/O command with a one in the bit position to be toggled. Note that it is possible to operate on multiple bits with the same byte in a single operation.

6.2.8 STATUS COMMANDS

The status commands allow the master node to access up to 256 bytes of memory on each slave device and slave device extension. These operations can be used to create a shared data structure between the master and slave nodes. Specific operation include read status and write status. Since these commands share a common format in the message data field, they are presented together.

The general data field format for status commands is shown in figure 37. This format allows a single command to operate on one or more locations with a single message. The data field is comprised of one or more pairs of bytes. The pairs include an address field and a data byte field. The address fields identify the locations at which the operation specified by the command is performed. The data byte field contains the data for the operation or acts as a place holder for the data that results from the operation.

Upon successful completion of a status command, a reply message is returned with an 00H response field and a data field with the same address fields as the order message and data byte fields as specified by the command. In all cases, the reply message length is equal to that of the order message.

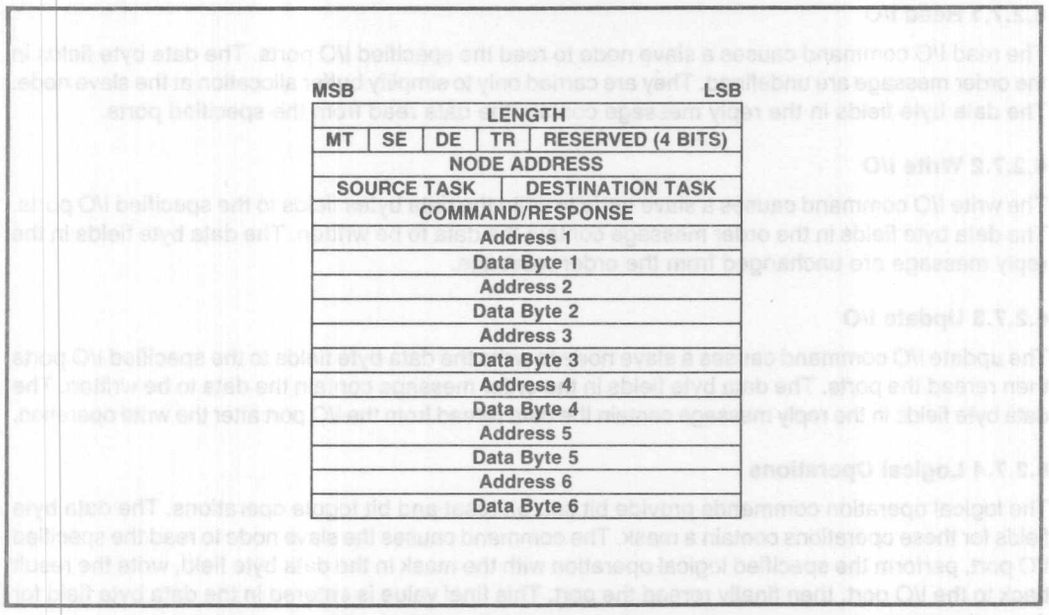


Figure 37. Data Field Format for Status Commands

6.2.8.1 Read Status

The read status command causes a slave node to read the specified memory locations. The data byte fields in the order message are undefined. They are carried only to simplify buffer allocation at the slave node. The data byte fields in the reply message contain the data read from the specified addresses.

6.2.8.2 Write Status

The write status command causes a slave node to write the data byte fields to the specified addresses. The data byte fields in the order message contain the data to be written. The data byte fields in the reply message are unchanged from the order message.

6.3 RAC RESPONSES

In general, RAC order messages expect RAC reply messages. The only cases where reply messages are not returned are for a successfully delivered reset slave order message or under certain error conditions as specified in section 5 (section 5 also specifies how to handle this case). When a reply message is returned, it indicates one of three cases: no error, message protocol error, or RAC error. Reply messages for the no error case are specified for each command in section 6.2. Reply messages for message protocol errors are specified in section 5. This section specifies the reply messages for the various RAC error conditions.

In general, RAC errors generate a reply message similar to message protocol errors. The reply message for RAC errors is simply the order message with the error response inserted in place of the command. the data field is returned intact. Table 7 summarizes the RAC error responses.

6.3.1 NO TASK

No task is generated in response to a delete task command in the case where the task to be deleted does not exist.

Table 7. RAC ERROR RESPONSES

RESPONSE	VALUE
No Task	80H
Task Overflow	81H
Register Bank Overflow	82H
Duplicate Function ID	83H
No Buffers	84H
RAC Protected	95H
Unknown RAC Command	96H

6.3.2 TASK OVERFLOW

Task overflow is generated in response to a create task command in the case where the slave device or slave device extension cannot support another task. This response indicates that it is necessary to delete a task before another can be created.

6.3.3 REGISTER BANK OVERFLOW

Register Bank Overflow is generated in response to a create task command that cannot be executed due to a lack of register bank resources.

6.3.4 DUPLICATE FUNCTION ID

Duplicate Function ID is generated in response to a create task command in the case where the task to be created has the same function ID (0FFH excepted) as a task currently recognized by the operating system.

6.3.5 NO BUFFERS

No buffers is generated in response to a create task command that cannot be executed due to a lack of memory resources.

6.3.6 RAC PROTECTED

RAC protected is generated in response to any remote access command when the RAC function is disabled. Before the remote access command can be recognized, an RAC protect command must be sent to enable the RAC function.

6.3.7 UNKNOWN RAC COMMAND

Unknown RAC command is generated in response to any RAC command that is not recognized. This response is used for both undefined and unimplemented commands.

7.0 MECHANICAL SPECIFICATION

The BITBUS interconnect provides mechanical specifications for cable connectors and a standard printed board. The cable connectors are used to connect cables to printed boards or cables to other cables. The standard printed board is specified to allow multiple manufacturers to build complementary products that fit into a consistent packaging scheme.

7.1 CONNECTOR SPECIFICATIONS

The BITBUS interconnect requires specification of two standard connectors: one for cable to printed board connection and one for cable to cable connections. Both connectors support the same signals. Four pins are used for the two differential signal pairs specified in section 2 of this specification for

the BITBUS interconnect. Four additional pins are provided for power distribution to low power nodes. Finally a high impedance ground (100 ohms to ground) is provided as specified in the RS485 specification (not required in all implementations). Figure 38 and 39 show the two connectors and table 8 lists the pin assignments.

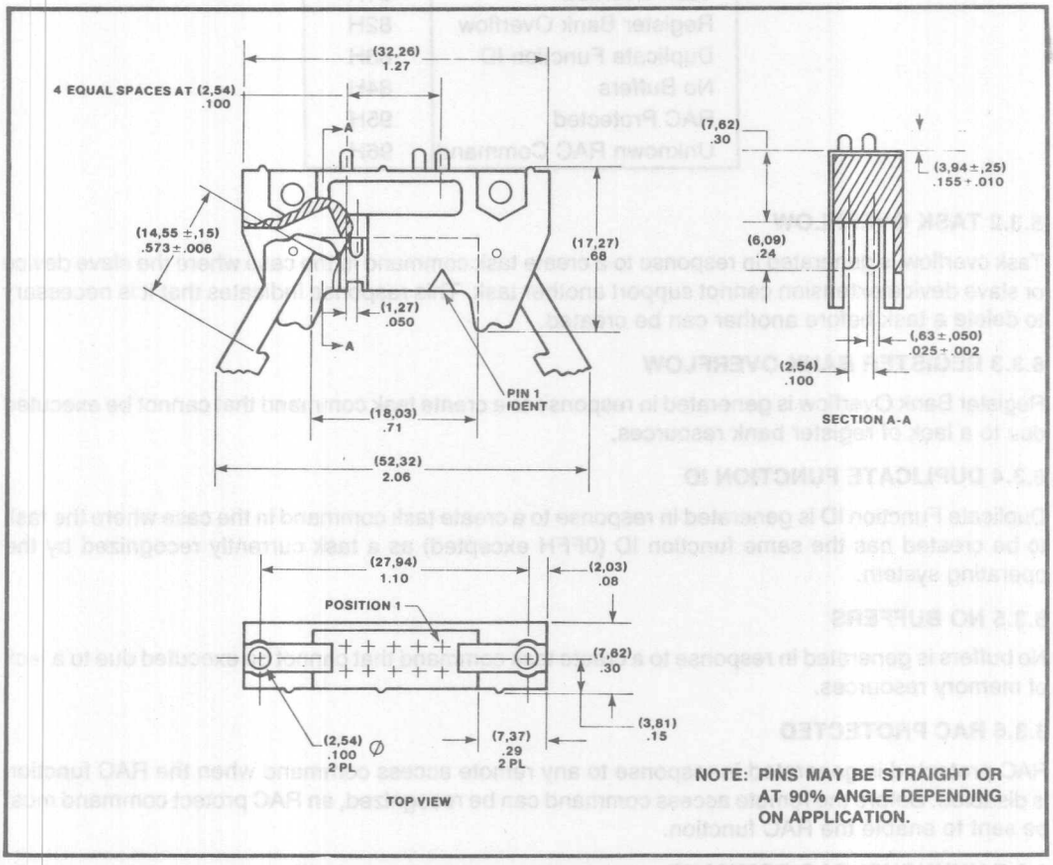


Figure 38. Printed Board Connector

Table 8. CONNECTOR PIN ASSIGNMENTS

SIGNAL	PRINTED BOARD CONNECTOR PIN #	CABLE CONNECTOR PIN #
+12V ±5%	1	1
+12V ±5%	2	6
GND	3	2
GND	4	7
DATA*	5	3
DATA	6	8
DCLK*/RTS*	7	4
DCLK/RTS	8	9
RGND (100 ohms to GND)	9,10	5

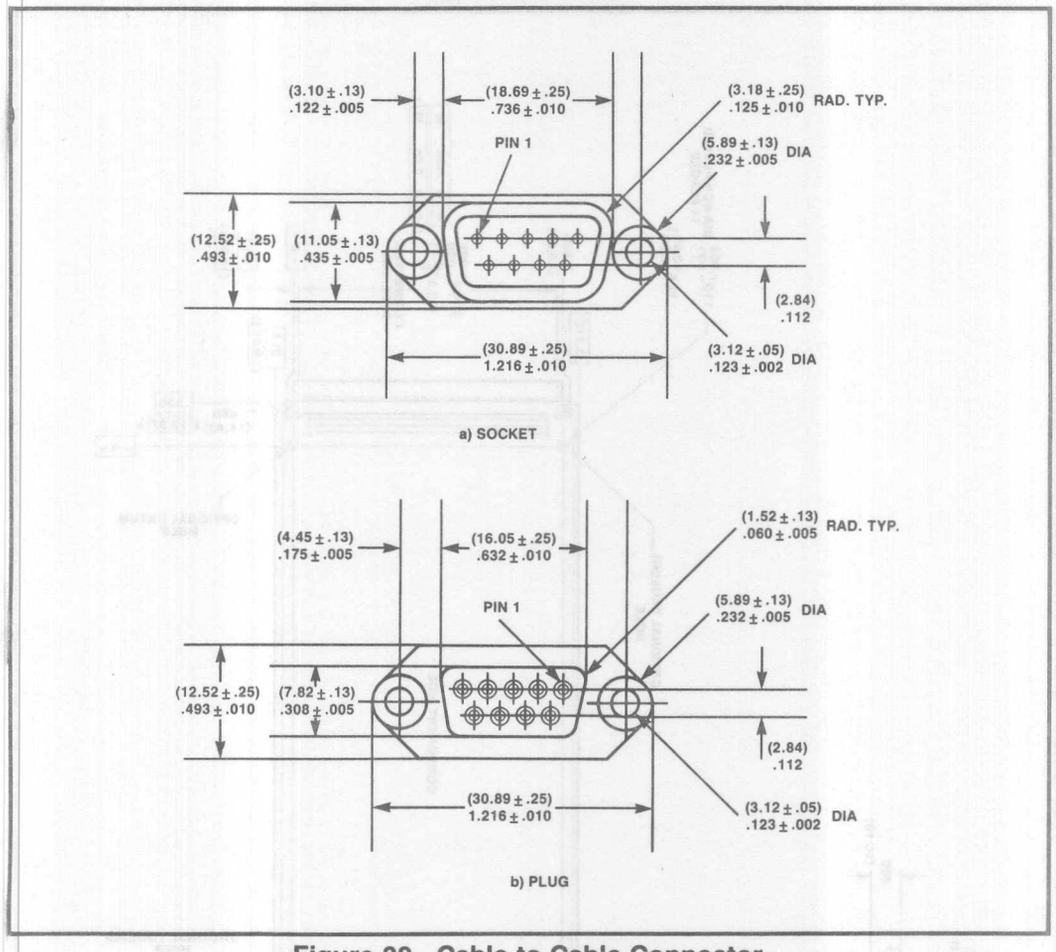


Figure 39. Cable to Cable Connector

The standard cable to printed board connector is a 10 pin latching pin and socket type with strain reliefs. This connector can support either flat cable or discrete wire connection.

The standard cable to cable connector is a 9 pin D-subminiature type. Versions of this connector are also available for flat cable or discrete wire connection.

7.2 I/O BOARD SPECIFICATION

The BITBUS I/O board is designed to be compatible with the single-high Multibus II board form factor. This definition is based on the International Electrotechnical Commission (IEC) standards. This section provides an overview of the mechanical dimensions of this board and presents standard pin assignments. For further details (e.g. board spacing, component height, ejector tabs, front panels, etc.) refer to the Mechanical Specification in the Multibus II Bus Architecture Specification.

7.2.1 BOARD DIMENSIONS

Figure 40 shows the board dimensions for the standard BITBUS I/O board. This board is a standard single-high, 220mm deep form factor.

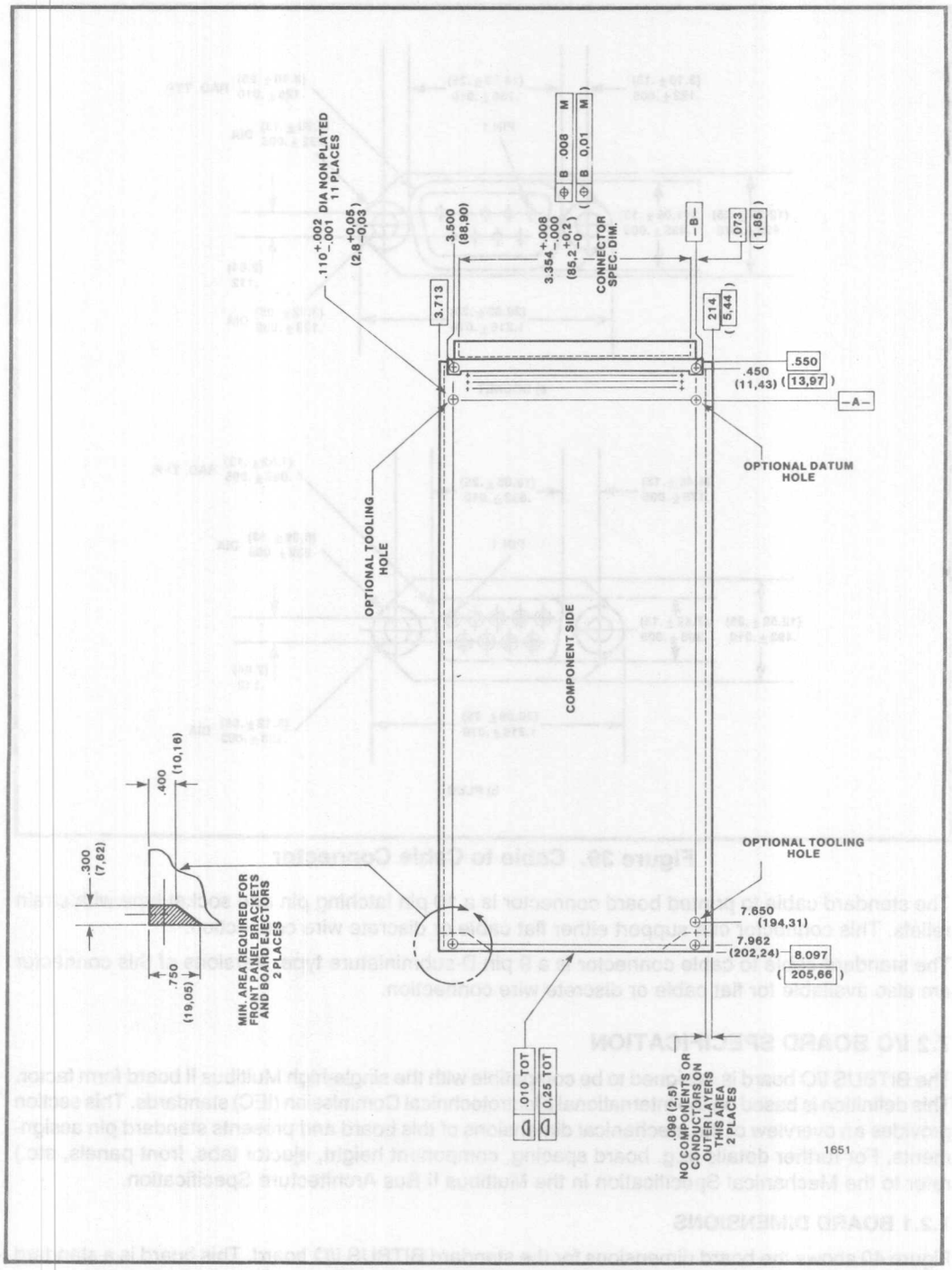


Figure 40. Standard Board Dimensions

7.2.2 CONNECTORS

The BITBUS I/O boards use two piece, 64-pin connectors. Figure 41 and 42 show the dimensional specification for the connectors. The right angle connectors on the printed board are IEC standard 603-2-IEC-C064-M; the receptacle connectors are IEC standard 603-2-IEC-C064-F. Figure 43 shows the relationship between connectors and boards in a subrack. Note that compatible recepticle connectors are available for flat cable, discrete wire or wire wrap connections in addition to the backplane version shown.

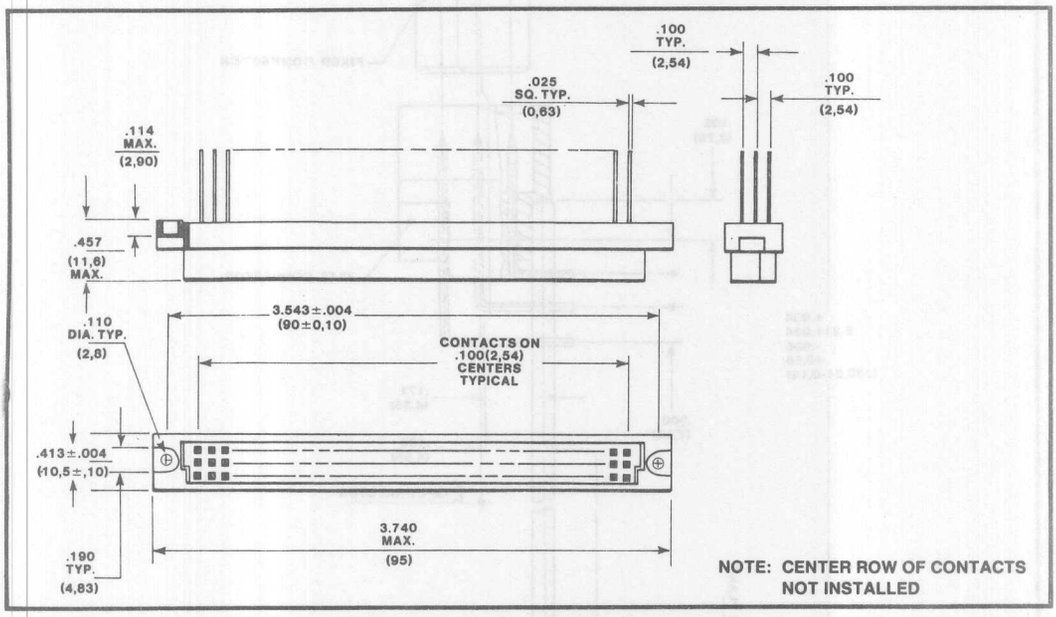


Figure 41. Board Connector

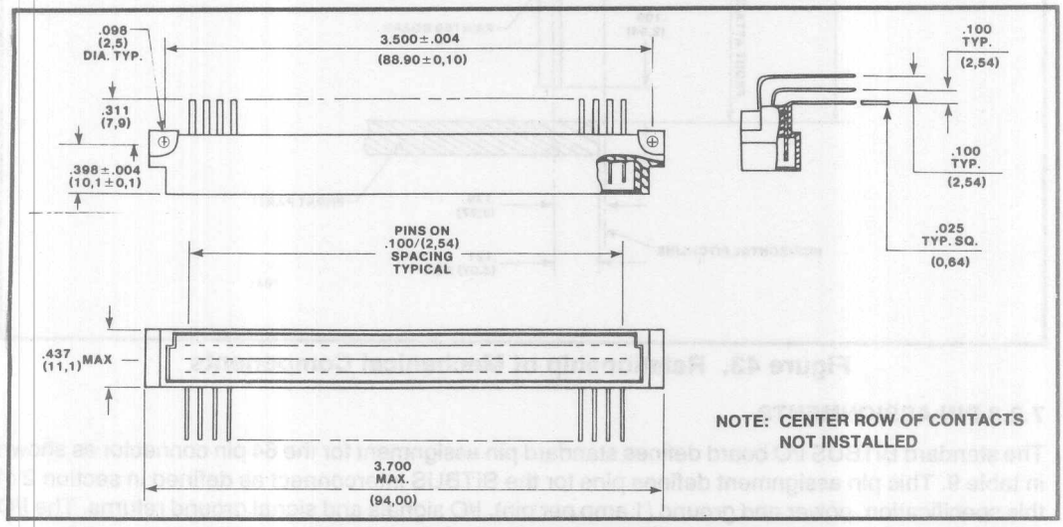


Figure 42. Backplane Connector

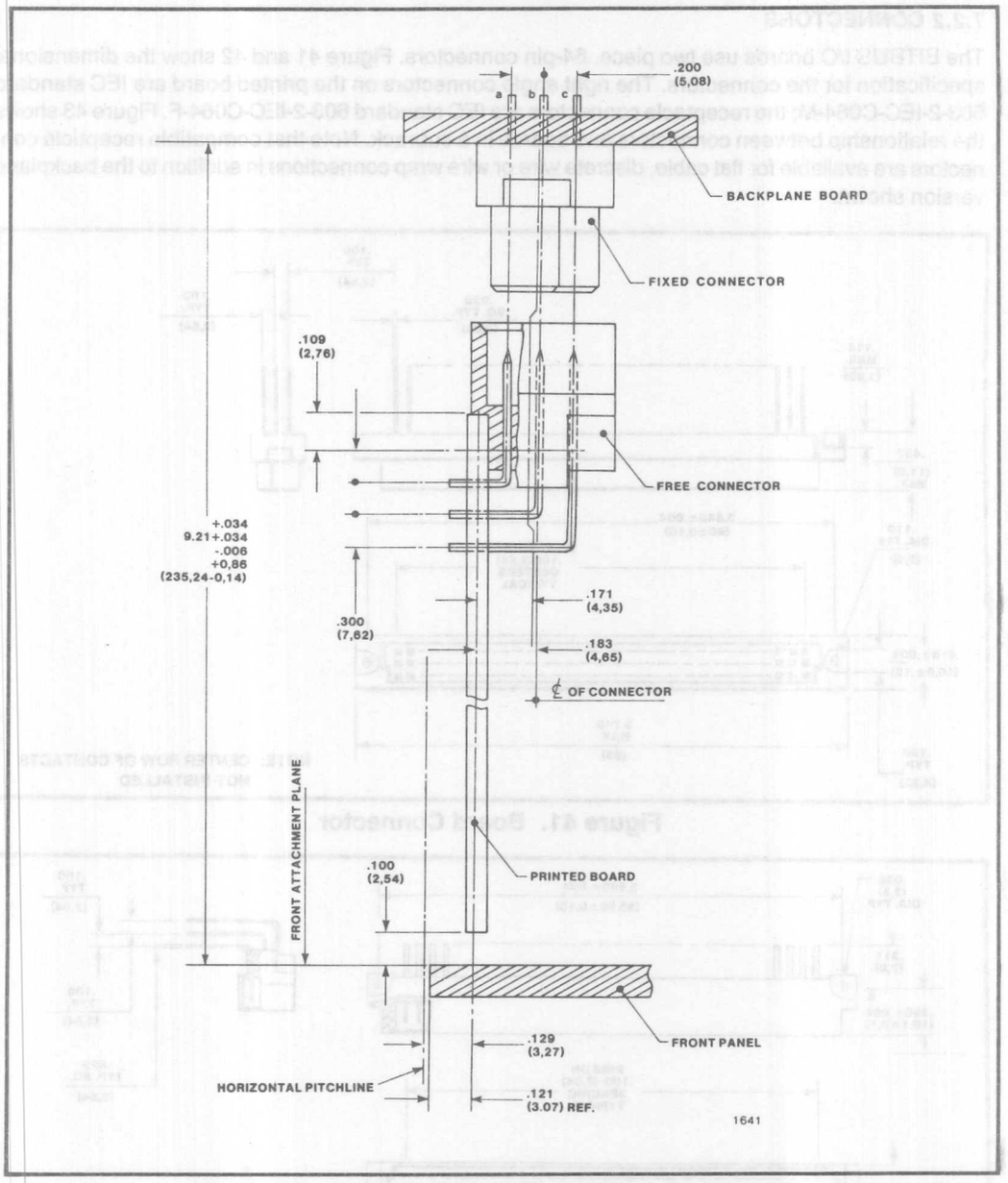


Figure 43. Relationship of Mechanical Components

7.2.3 PIN ASSIGNMENTS

The standard BITBUS I/O board defines standard pin assignment for the 64 pin connector as shown in table 9. This pin assignment defines pins for the BITBUS interconnect as defined in section 2 of this specification, power and ground (1 amp per pin), I/O signals and signal ground returns. The I/O signal lines are defined as required by each implementation.

Table 9. I/O BOARD PIN ASSIGNMENT

PIN #	SIGNAL	PIN #	SIGNAL
1a	GND	1c	GND
2a	+5V	2c	+5V
3a	DATA	3c	DATA*
4a	DLCK/RTS	4c	DLCK*/RTS*
5a	I/O	5c	RGND
6a	I/O	6c	I/O
7a	I/O	7c	GND
8a	I/O	8c	I/O
9a	I/O	9c	GND
10a	I/O	10c	I/O
11a	I/O	11c	GND
12a	I/O	12c	I/O
13a	I/O	13c	GND
14a	I/O	14c	I/O
15a	I/O	15c	GND
16a	I/O	16c	I/O
17a	I/O	17c	GND
18a	I/O	18c	I/O
19a	I/O	19c	GND
20a	I/O	20c	I/O
21a	I/O	21c	GND
22a	I/O	22c	I/O
23a	I/O	23c	GND
24a	I/O	24c	I/O
25a	I/O	25c	GND
26a	I/O	26c	I/O
27a	I/O	27c	GND
28a	I/O	28c	I/O
29a	I/O	29c	GND
30a	+12V	30c	−12V
31a	+5V	31c	+5V
32a	GND	32c	GND

8.0 LEVELS OF COMPLIANCE

The BITBUS interconnect specification contains many features that are required for compatibility and other features that are only required for additional capabilities. This section clarifies the various levels of compliance that may be supported by various implementations of the BITBUS interconnect. This information is included to allow use of the BITBUS interconnect products of varying capabilities manufactured by diverse vendors.

8.1 CONCEPT OF LEVEL OF COMPLIANCE

The concept of level of compliance is provided to guarantee compatibility of BITBUS interface products that support the specification at different levels. The basic rule is that all products are required

to support a minimum level of the specification. This results in a minimal level at which all BITBUS interconnect products can interact. Furthermore, enhancements beyond this minimal level are controlled. This results in a guaranteed compatibility between a group of products that all support the same enhanced capabilities.

8.2 COMPLIANCE LEVELS

This section presents the various levels of compliance for each level of the specification. Each capability supported beyond the minimum shall be documented in the product data sheet.

8.2.1 ELECTRICAL INTERFACE

The electrical interface supports variability in bit rates, load characteristics, and repeater characteristics. The bit rates specified in section 2 include 2.4 mbits/sec synchronous and 375 kbits/sec and 62.5 kbits/sec self clocked. Of these only 375 kbits/sec is required. The others may be optionally supported. The load for a BITBUS node shall be specified in terms of the standard load defined in section 3.2.1. Repeaters are specified in terms of the standard repeater delays defined in section 3.3.5.

8.2.2 DATA LINK PROTOCOL

The data link protocol has no optional features. The full protocol shall be supported for compatibility.

8.2.3 MESSAGE PROTOCOL

The message protocol supports variability in message length. All implementation shall support the standard 5 byte header and 13 byte data field. Implementation may optionally support larger data fields.

8.2.4 REMOTE ACCESS AND CONTROL

Support of the remote access and control function is completely optional. The only requirement is that supported commands shall meet the specification.

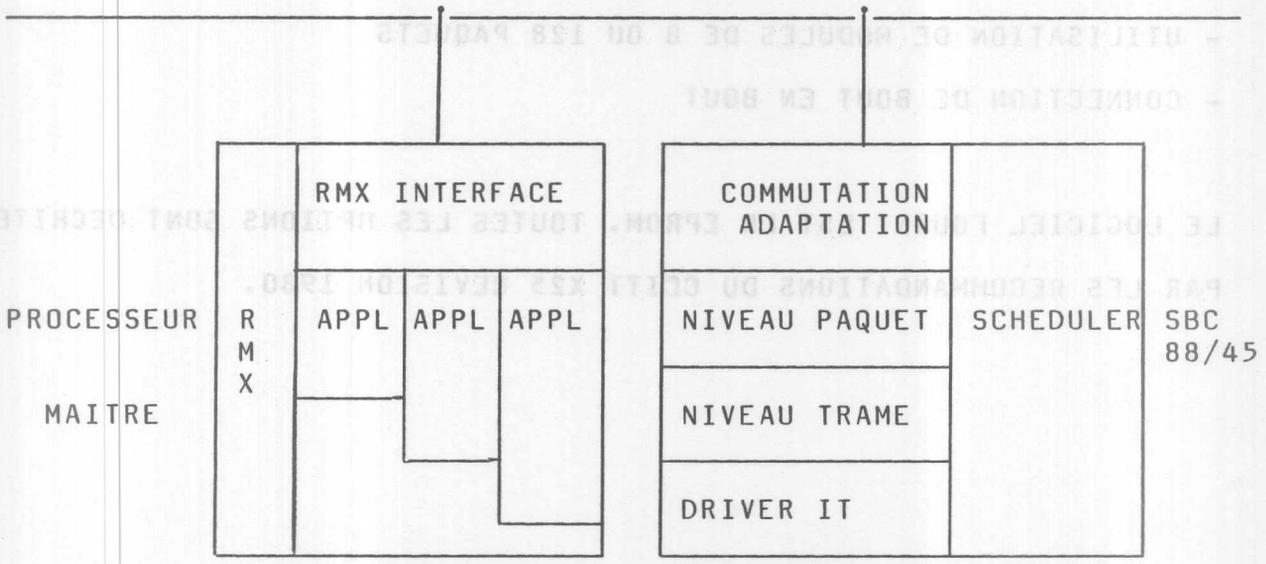
8.2.5 MECHANICAL

The mechanical specification contains the option of supporting the standard I/O board specification. As with RAC, the only requirement is that if supported, it shall meet the specification.


- 5 -
LOGICIEL X25
T.I.T.N

EN MATIERE DE COMMUNICATION CONCERNANT X25, INTEL A RETENU LA SOCIETE T.I.T.N.
CE PRODUIT LOGICIEL APPARAIT DANS LES "YELLOW PAGES" ET METROLOGIE LE DISTRIBUE EN FRANCE.
LE LOGICIEL X25 DE T.I.T.N. EST PORTABLE SUR TOUT OPERATING SYSTEM. CEPENDANT T.I.T.N. PROPOSE UNE SOLUTION PRECONFIGUREE POUR LA CARTE SBC 88/45 AVEC UNE INTERFACE L'ASSOCIANT A iRMX 86 DANS CE CAS L'INTERFACE EST BASEE SUR UNE FILE DE COMMANDES.

L'ARCHITECTURE EST LA SUIVANTE :



LE LOGICIEL X25 DE T.I.T.N. EST ENTIEREMENT CONFIGURABLE.

LES PARAMETRES SONT DEFINIS A L'INITIALISATION :

- NOMBRE DE LIGNES PHYSIQUES
- TYPE ET CARACTERISTIQUES DES TRAVAUX
- NOMBRE ET TYPE DE CIRCUIT VIRTUEL POUR CHAQUE LIGNE
- UTILISATION DE LA LIGNE SYSTEME
- UTILISATION DE MODULES DE 8 OU 128 PAQUETS
- CONNECTION DE BOUT EN BOUT

LE LOGICIEL FOURNI EST EN EPROM. TOUTES LES OPTIONS SONT DECRITES PAR LES RECOMMANDATIONS DU CCITT X25 REVISION 1980.

I - LES LOGICIELS DE COMMUNICATION PORTABLES THOMSON-TITN

I-1- PRESENTATION GENERALE

THOMSON-TITN propose une gamme complète de logiciels X.25 et TELETEX compatibles avec les dernières recommendations du CCITT :

- Logiciels X.25 permettant l'accès à tous les réseaux X.25 mondiaux de commutation de paquets.

- Logiciel X.29 permettant la réalisation de serveurs accessibles par des terminaux asynchrones connectés au réseau à travers un PAD.

- Logiciel d'émulation PAD X.28 - X.3 permettant d'accéder à un serveur PAD en France ou à l'étranger avec une liaison securisée.

- Logiciel TELETEX permettant la connexion au service TELETEX de transmission de documents à travers un réseau de commutation de paquets X.25, de commutation de circuits X.21 ou téléphonique commuté avec l'option LAPX.

- Logiciel X.21 d'accès au réseau de commutation de circuits type TELECOM 1 ou RTC 64.

Les logiciels de commutation portables THOMSON-TITN, écrits en "C", peuvent être implantés sur la plupart des microprocesseurs et sur de nombreux minicalculateurs.

I-2- PORTAGE DU LOGICIEL

THOMSON-TITN assume la responsabilité complète du portage de ses logiciels de communication dans l'environnement matériel du client.

Le portage comprend l'implémentation du logiciel sur le matériel du client et l'adaptation aux interfaces matériels. Un interface utilisateur standard de type entrées-sorties est adapté au système d'exploitation du client.

Des spécifications détaillées d'implémentation sont fournies au client pour approbation dans un délai de deux mois après la commande.

I-3- SPECIFICATIONS MATERIELLES MINIMALES

Les logiciels de communication portables THOMSON-TITN utilisent généralement un coupleur de communication intelligent comprenant :

- un processeur pour lequel un compilateur "C" est disponible.
- 64 KO de mémoire dont au minimum 16 KO de RAM.
- 1 timer avec un compteur pour les délais logiciels, générant une interruption toutes les dix millisecondes.
- 1 SIO synchrone programmable en HDLC couplé en mode programme ou en DMA.
- 1 SIO asynchrone pour le moniteur de ligne optionnel.
- 1 interface de type "boite à lettre" avec le processeur principal.

Les logiciels X.25, X.29 et X.21 sont implantés sur le coupleur de communications.

Les logiciels Télétex et PAD sont implantés sur le processeur principal du fait de leur forte interaction avec le système d'exploitation.

THOMSON-TITN peut étudier le portage du logiciel sur tout autre type de configuration matérielle. Des implémentations sur des systèmes monoprocesseur voir monotaches sont possible.

I-4- GENERATION DYNAMIQUE

Tous les logiciels de communications THOMSON-TITN sont entièrement paramètrables à l'initialisation.

Les paramètres sont définis à l'initialisation du logiciel qui construit automatiquement ses tables et buffers dans un segment de données défini par l'utilisateur.

Ces paramètres fixent le dimensionnement et les caractéristiques de l'abonnement ainsi que les spécificités de chaque réseau.

Certains paramètres peuvent être rendus inaccessibles à l'utilisateur final.

I-5- INCIDENTS MATERIELS

Tout incident matériel détecté par le logiciel peut provoquer un branchement dans une séquence de test fournie par le client (par défaut arrêt du logiciel).

II - LOGICIEL X.25

II-1- PRESENTATION

Le logiciel X.25 THOMSON-TITN permet la connexion à tous les réseaux X.25 de commutation de paquets européens et mondiaux.

Le X.25 est maintenant reconnu comme un standard mondial pour les transmissions de données. Le nombre de réseaux X.25 dans le monde est en croissance rapide.

Le logiciel portable X.25 THOMSON-TITN est un moyen rapide et sûr d'implanter un interface X.25 sur un matériel quelconque. Il est entièrement écrit en langage "C".

Il est certifié ou en cours de certification sur les principaux réseaux mondiaux :

- TRANSPAC	(France)	- DATEX-P	(Allemagne)
- EURONET	(Europe)	- PSS	(Grande Bretagne)
- TYMNET	(U.S.A.)	- KDD	(Japon)
- TELENET	(U.S.A.)	- DCS	(Belgique)
- UNINET	(U.S.A.)	- DATAPAK	(Canada)

II-2- SERVICES SUPPORTES

Le logiciel X.25 supporte l'ensemble des services définis par les dernières recommendations du CCITT :

- circuits virtuels permanents ou commutés,
- négociation des paramètres de contrôle de flux (taille de fenêtre, taille de paquet),
- classe de débit
- numérotation étendue à 128 (liaisons satellites)
- spécialisation des voies logiques en émission ou réception,
- groupes fermés d'abonnés
- sélection rapide
- paquet REJ
- bits Q et M
- bit D
- connexion spécialisées ETTD - ETTD
- LAP et LAPB
- extension d'adresse
- etc...

Le service "datagrammes" n'est supporté actuellement par aucun réseau public et n'est pas implanté.

II-3- IMPLEMENTATION DU LOGICIEL

Le logiciel X.25 est, en général, implanté sur un coupleur de communication intelligent indépendant. Il s'interface avec l'application à travers des "boites à lettres" en mémoire commune.

Les programmes de téléchargement, d'initialisation et d'interface utilisateur sur le processeur principal sont fournis par THOMSON-TITN.

La taille du logiciel est de environ 40 K octets. La taille M des tables et buffers varie avec les caractéristiques de la ligne X.25 :

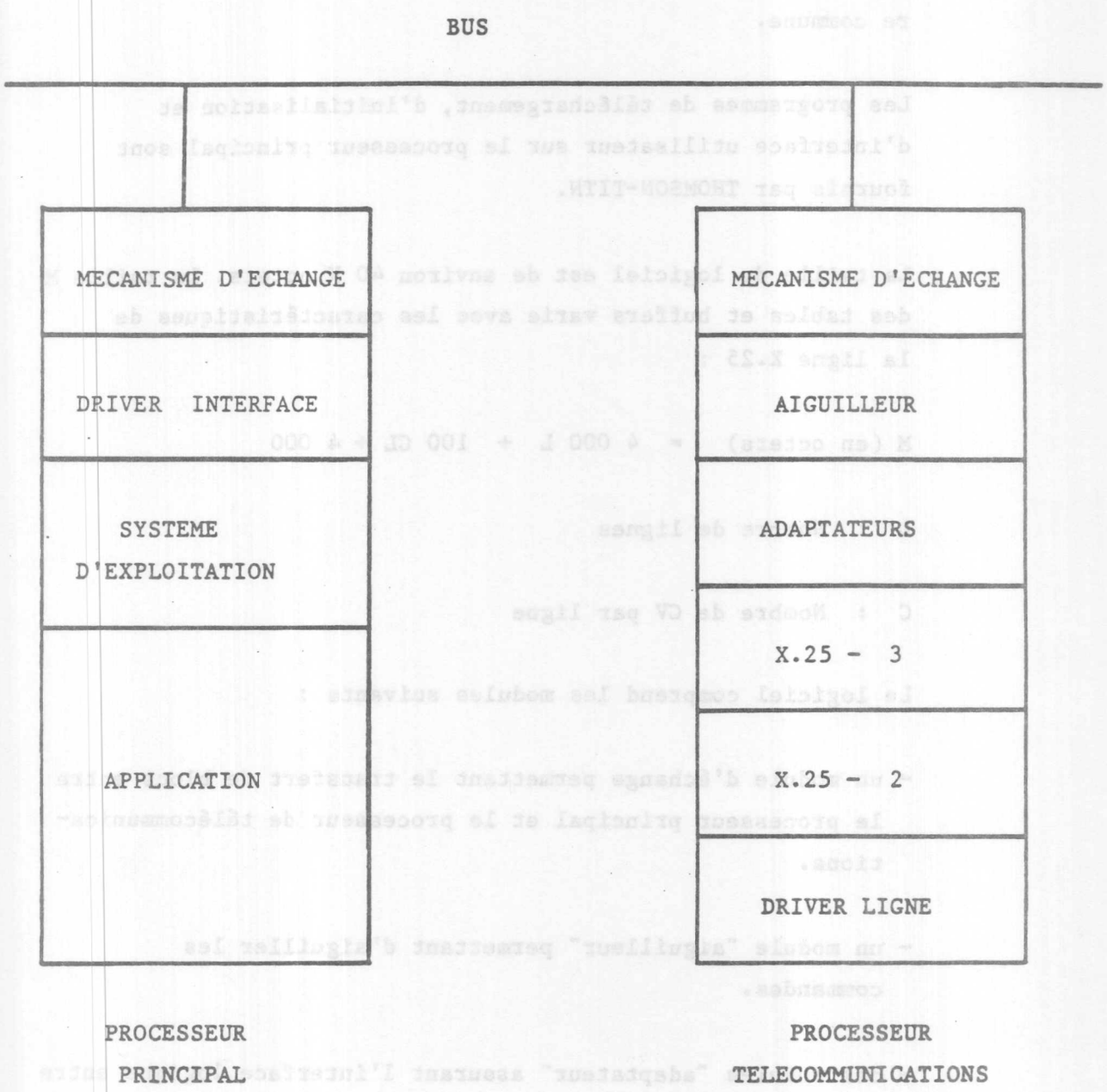
M (en octets) = 4 000 L + 100 CL + 4 000

L : Nombre de lignes

C : Nombre de CV par ligne

Le logiciel comprend les modules suivants :

- un module d'échange permettant le transfert de blocs entre le processeur principal et le processeur de télécommunications.
- un module "aiguilleur" permettant d'aiguiller les commandes.
- des modules "adaptateur" assurant l'interface logique entre le driver de l'application et le logiciel X.25.
- un module paquet X.25 niveau 3
- un module trame X.25 niveau 2
- le driver ligne gérant le boîtier série d'Entrée/Sortie.


BUS
MECANISME D'ECHANGE
DRIVER INTERFACE
SYSTEME D'EXPLOITATION
APPLICATION
MECANISME D'ECHANGE
AIGUILLEUR
ADAPTATEURS
X.25 - 3
X.25 - 2
DRIVER LIGNE
PROCESSEUR PRINCIPAL
PROCESSEUR TELECOMMUNICATIONS

II-4- INTERFACE X.25

L'interface X.25 permet à l'application de dialoguer avec le logiciel X.25. Il comprend trois niveaux :

- le "mécanisme d'échange" qui définit des boites à lettres utilisées pour l'échange de blocs de commande entre le X.25 et l'interface utilisateur à travers une mémoire double accès avec un mécanisme de synchronisation.

- l'interface "driver" qui définit un interface de type "entrées-sorties" avec des ordres classiques (OPEN, CLOSE, READ, WRITE, IOCTL).

- l'interface du logiciel X.25 (appels sortants, échange de paquets, appels entrants, etc...). Les commandes sont multiplexées par des unités logiques vers les différents circuits virtuels.

Le X.25 peut avoir deux modes de fonctionnement :

- le "mode paquet" où l'application échange avec le X.25 des paquets de données au format standard X.25. Ce mode donne l'accès à l'ensemble des services X.25. Il nécessite une certaine connaissance du X.25.

- le "mode données" ou "mode transparent" où l'application dialogue avec le X.25 avec un interface standard d'entrées-sorties. Le X.25 se charge du découpage des blocs en paquets. L'utilisateur se contente d'effectuer des "READ" et des "WRITE". Ce mode restreint les services accessibles du X.25 mais permet d'interfacer des applications standards.

L'utilisateur peut commuter d'un mode à l'autre sur un circuit virtuel en cours de connexion.

Les évènements entrants asynchrones (appel entrant, incident ligne) sont envoyés à un processeur de contrôle fourni par l'utilisateur.

II-5- PERFORMANCES

Le X.25 peut gérer environ 60 paquets entrants ou sortants par seconde avec un SIO couplé en accès direct mémoire. Ces performances peuvent varier avec la puissance du processeur et la qualité du compilateur.

II-6- PARAMETRES D'INITIALISATION

Les paramètres d'initialisation du logiciel comprennent toutes les caractéristiques de l'abonnement :

- réseau utilisé
- taille maximale de paquet
- vitesse de ligne
- caractéristiques de l'abonnement
- dimensionnement
- temporisation
- etc...

Le lancement du logiciel s'effectue en trois phases :

- chargement du logiciel (une version en REPROM est aussi disponible)

- initialisation des paramètres

- lancement du logiciel dans la séquence d'initialisation.

II-7- MONITEUR DE LIGNES (LMS)

Le moniteur de ligne optionnel permet la trace complète ou selective du trafic échangé avec la liaison X.25 sur un terminal asynchrone de type écran. Il permet d'assurer une mise au point ou un diagnostic local ou distant.

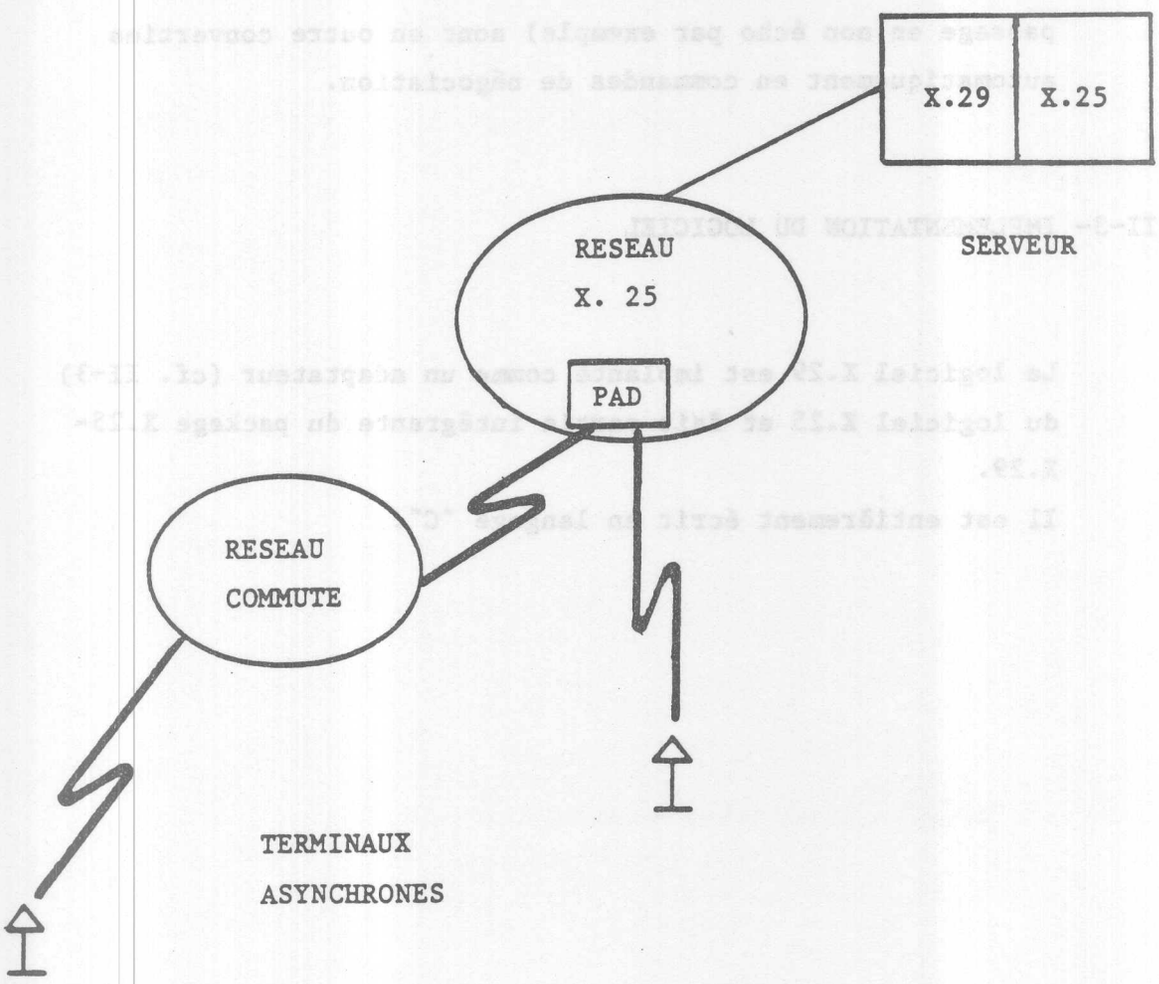
Le logiciel peut tracer des paquets ou des trames d'un type donné ou sur une voie logique donnée. Les critères de sélection sont fixés à l'initialisation. Une version interactive sera disponible prochainement.

Les entêtes des paquets ou trames sont affichées sous forme symbolique. Les autres données sont affichées en héxadécimal.

III - LOGICIEL X.29

III-1- PRESENTATION GENERALE

Le logiciel X.29 permet de réaliser un serveur sur un réseau X.25 accessible par un terminal asynchrone quelconque du réseau connecté à travers un PAD.



III-2- SERVICES SUPPORTES

Le logiciel X.29 assure la fragmentation réassemblage des lignes. les données sont passées ligne par ligne à l'application.

Les commandes de négociation peuvent être passées au logiciel X.29 dans le format standard CCITT.

Des requêtes standard du système d'exploitation (du type passage en non écho par exemple) sont en outre converties automatiquement en commandes de négociation.

III-3- IMPLEMENTATION DU LOGICIEL

Le logiciel X.29 est implanté comme un adaptateur (cf. II-3) du logiciel X.25 et fait partie intégrante du package X.25-X.29.
Il est entièrement écrit en langage "C".

III-4- INTERFACE UTILISATEUR

L'interface X.29 est analogue a l'interface X.25.

En mode "transparent", après l'établissement de la connexion, l'application peut gérer un terminal PAD de la même manière qu'un terminal local (avec quelques restrictions mineures dûes au PAD).

Un circuit virtuel est dynamiquement assigné à X.29 au moment de son établissement.

III-5- PERFORMANCES

Le logiciel X.25 - X.29 peut gérer environ 40 paquets par seconde (avec un SIO en accès direct mémoire).

III-6- PARAMETRES D'INITIALISATION

Le X.29 ne possède aucun paramètre spécifique par rapport au X.25.

IV - LOGICIEL X28 - X3

IV - 1 PRESENTATION GENERALE

Le logiciel X.28-X3 permet d'emuler un terminal connecté à un PAD avec un calculateur connecté directement en X25 à un réseau.

Ce logiciel permet :

- Une liaison fiabilisée en supprimant la connexion non sécurisée entre le terminal asynchrone et le PAD.

- Une performance supérieure en supprimant la limite actuelme de 1200 bauds sur les accès PAD.

- La facilité de connexions internationales qui sont très restreintes avec un terminal PAD pour des raisons de taxation.

Il nécessite l'utilisation du logiciel TITN X.25.

IV - 2 SERVICE SUPPORTES

Le logiciel X.28-X3 assure l'interface avec le ou les terminaux asynchrones du système et la fragmentation réassemblage des caractère en paquets.

Il assure la négociation locale ou distante des paramètres.

IV - 3 IMPLEMENTATION DU LOGICIEL

Le logiciel X.28-X3 doit être implanté sur le processeur gérant les terminaux sous le système d'exploitation du client.

Il est entièrement écrit en langage "C".

Il est considéré comme un programme d'application vis-à-vis du système d'exploitation.

IV - 4 INTERFACE UTILISATEURS

Le X.28-X3 n'a pas d'interface de type "application" car il interface directement les terminaux, à travers le système d'Entrées/Sorties, et le logiciel X.25.

Le dialogue opérateur est analogue à celui d'un terminal PAD.

Une sélection de paramètres par "menu" peut être proposée dans le cadre du portage du logiciel.

V - LOGICIEL TELETEX

V - 1 PRESENTATION GENERALE

Le logiciel TELETEX permet de connecter un calculateur au service TELETEX à travers les réseaux X.25 de commutation de paquets, X.21 à commutation de circuits, ou téléphonique commuté.

Il comprend les couches transport (CCITT S70) et session (CCITT S62) du protocole TELETEX. Il utilise le logiciel TITN X.25.

Les extensions X.21 pour l'accès aux réseaux circuits (Allemagne, Suisse, Scandinavie, Canada) et LAPX pour l'accès au réseau téléphonique commuté (France) sont intégrés au logiciel X.25.

Ce logiciel est conforme aux recommendations des administrations des différents pays concernant leur service TELETEX présent ou futur (France, Allemagne, Canada, Etats-Unis).

V - 2 SERVICES SUPPORTES

Le logiciel TELETEX supporte tous les services du Télétex de base définies par le CCITT.

Les extensions des services TELETEX en cours de définition seront implantés dans le logiciel à une date ultérieure :

- Multiplexage au niveau transport.
- Transmission bidirectionnelle simultanée.
- Extensions graphiques (CCITT Sa)..

Le logiciel gère un journal des documents entrants et sortants sur un fichier disque ou une imprimante.

V - 3 CARACTERISTIQUES MINIMALES DU MATERIEL ET DU LOGICIEL DE BASE

En plus des caractèristiques liées au logiciel X.25 (cf I-3), le logiciel TELETEX nécessite une mémoire de masse et environ 40 KO de mémoire sur le processeur central.

Le logiciel TELETEX utilise un scheduler de tâches interne et peut donc est implanté sur tout système d'exploitation mono ou multitâhce possédant un système de gestion de fichiers.

V - 4 IMPLEMENTATION DU LOGICIEL

Le logiciel TELETEX est implanté comme un programme d'application s'interfaçant avec le logiciel X.25 TITN.

Il est entièrement écrit en langage "C".

V - 5 INTERFACE

Les documents sortants doivent être générés par un logiciel de traitement de texte du client et soumis au logiciel TELETEX en format S61 à travers un fichier disque.

Les documents reçus sont fournis à l'utilisateur en format S61 TELETEX à travers un fichier disque.

Chaque document est précédé d'une entête contenant les caractèristiques d'acheminement de ce document.

Le logiciel TELETEX est entièrement paramétrable à l'initialisation comme le logiciel X.25.

VI - LOGICIEL X21

Le logiciel X.21 peut être ajouté au logiciel X.25 pour permettre l'établissement de communication en point à point sur un réseau X.21 à commutation de circuits (TELECOM1, ou RTC.64).

Ce logiciel nécessite un interface V24 standard auquel est ajouté les deux signaux parallèles C et I du standard X.21.

Il gère :

- La phase d'établissement et de rupture d'appel conformément à l'Avis X.21 du CCITT.
- Les services complémentaires prévus dans les START d'interface X.21 de TELECOM1, à l'exclusion du multipoint.

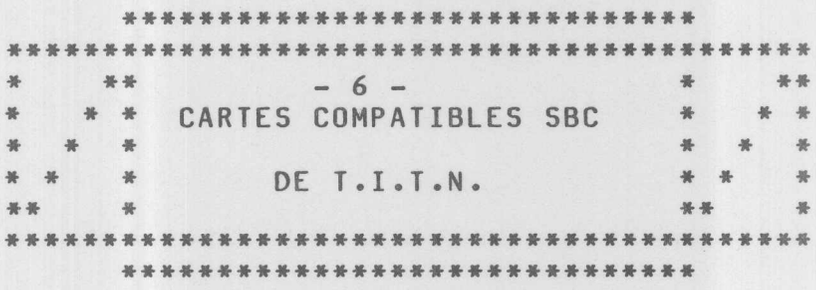
Le logiciel est principalement implanté au niveau du driver ligne.

Un processus indépendant gère les services complémentaires TELECOM1.

Il est presque intégralement écrit en langage "C".

Le logiciel X.21 est en général intégré au logiciel X.25 pour permettre une communication ETTD - ETTD X.25 après l'établissement du circuit.

Une version du logiciel indépendante, permettant de réaliser un interface X.21 sans utilisation du X.25 sera aussi disponible.


- 6 -
CARTES COMPATIBLES SBC
DE T.I.T.N.

CARTES COMPATIBLES SBC DE T.I.I.M.

ENTRE AUTRES ACTIVITES LA SOCIETE T.I.T.N. PROPOSE DES CARTES COMPATIBLES SBC, CELLES-CI SONT DISTRIBUEES PAR METROLOGIE.

AU CATALOGUE DE T.I.T.N. IL CONVIENT DE MENTIONNER L'EXISTENCE DE CARTES UNITES CENTRALES ORGANISEES SOIT AUTOUR DU 68B09, DU 68000 OU DU 8086. LES PRINCIPALES FONCTIONNALITES DE CES CARTES SONT :

TYPE	TN 68/31	TN 68/35	TN 68/44	TN 68K30	TN 86/40
CPU	68B09	68B09	68B09	68000	8086
VITESSE	1,8 MHZ	1,8 MHZ	1,8 MHZ	7,5 MHZ	5-8 MHZ
MEMOIRE RAM	16 KO	64 KO	16 KO	128 KO	32-128 KO
MEMOIRE EPROM	32 KO	32 KO	32 KO	64 KO	32 KO
LIAISON SERIE	3xRS232 DONT : 2xRS422	2xRS232 OU RS422	4xRS232	2xRS232	2xRS422 1xRS232
I/O PARALLELE	NON	48	8	NON	24
SBX	NON	NON	NON	1	1

POUR OBTENIR DES INFORMATIONS COMPLEMENTAIRES, NOUS CONTACTER...

PAR AILLEURS T.I.T.N REALISE D'AUTRES CARTES POUR LESQUELLES UNE SPECIFICATION ABREGEE VOUS EST COMMUNIQUEE CI-APRES :

IL S'AGIT DES CARTES :

- TN 03/TN 04 : CARTES PERMETTANT DE REPETER LE MULTIBUS D'UN BAC A CARTES A UN AUTRE BAC A CARTES.

- TN 68/39 : C'EST UNE CARTE D'EXTENSION DE 8 LIGNES SERIE RS232.

- TN 68/18 : C'EST UNE CARTE DE 8 KO DE RAM SECOURUE PAR BATTERIE AVEC EN PLUS L'ENTRETIEN DE L'HEURE.

- TNBX 05 : C'EST UN MULTIMODULE SBX DOTE DE 2 PORTS SERIE MULTI-PROTOCOLES.

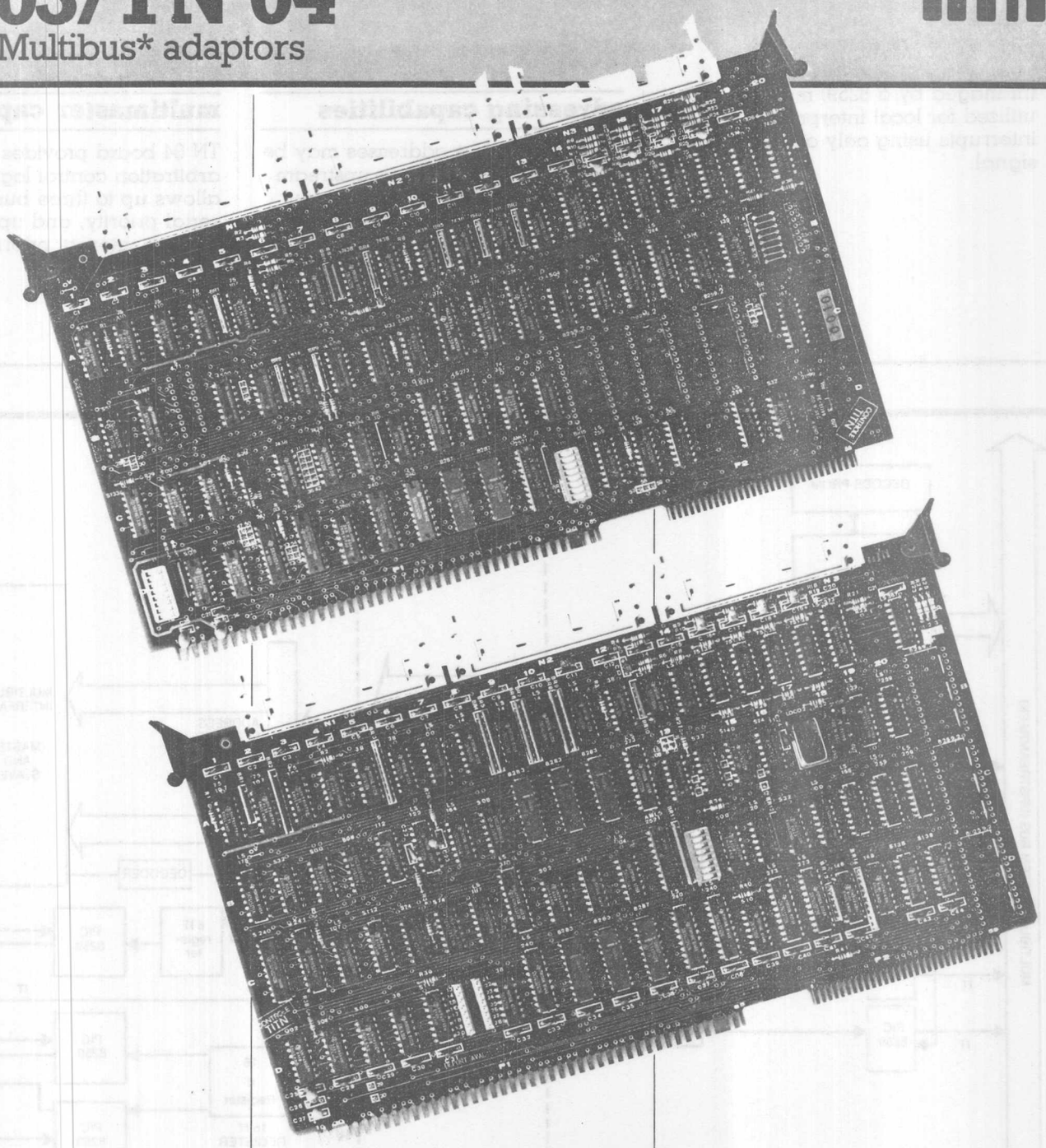
- TN 68100 : C'EST UNE CARTE MEMOIRE IMAGE GRAPHIQUE FAITE POUR ETRE ASSOCIEE A UN MONITEUR NOIR ET BLANC 625 LIGNES ENTRELACEES A 50 HZ
LA DEFINITION DE L'IMAGE VISUALISEE EST DE 864 x 572 POINTS TANDIS QUE L'IMAGE MEMORISEE EST DE 864 x 1144 POINTS.

- TN 68110 : C'EST UNE CARTE MEMOIRE IMAGE GRAPHIQUE FAITE POUR ETRE ASSOCIEE A UN MONITEUR COULEUR.
LA RESOLUTION EST DE 256 x 256 POINTS AVEC UNE PROFONDEUR DE 6 BITS PAR PIXEL.

TN 03/TN 04

multiple Multibus* adaptors

TITN



functional description

These boards provide the interfacing of several MULTIBUS* racks.

The upstream board acts as the master in the exchange between boards of different racks.

However there are no limitations on the control of the MULTIBUS* in the case of exchange between boards of the same rack.

The fields of MULTIBUS* addresses are "prom-fused" in the upstream adaptor.

These address fields are used for memory and I/O transfers in the upstream rack dedicated to the exchanges with the other rack.

These boards provide data registers and Time-out logic, so that the bus occupation is optimized in the two racks.

interrupt capability

18 levels of interrupts are available in two ways for interrack interrupts.

- 2 Multibus* interrupts are directly transmitted.
- 16 levels of vectored interrupts are dedicated to interprocessor interrupts between boards distributed in separate racks; they are managed by 2 PIC 8259's.

In each rack, 8 other interrupt levels

Trademark from Intel corporation.

(managed by a 8259) may be utilized for local interprocessor interrupts using only one Multibus* signal.

adressing capabilities

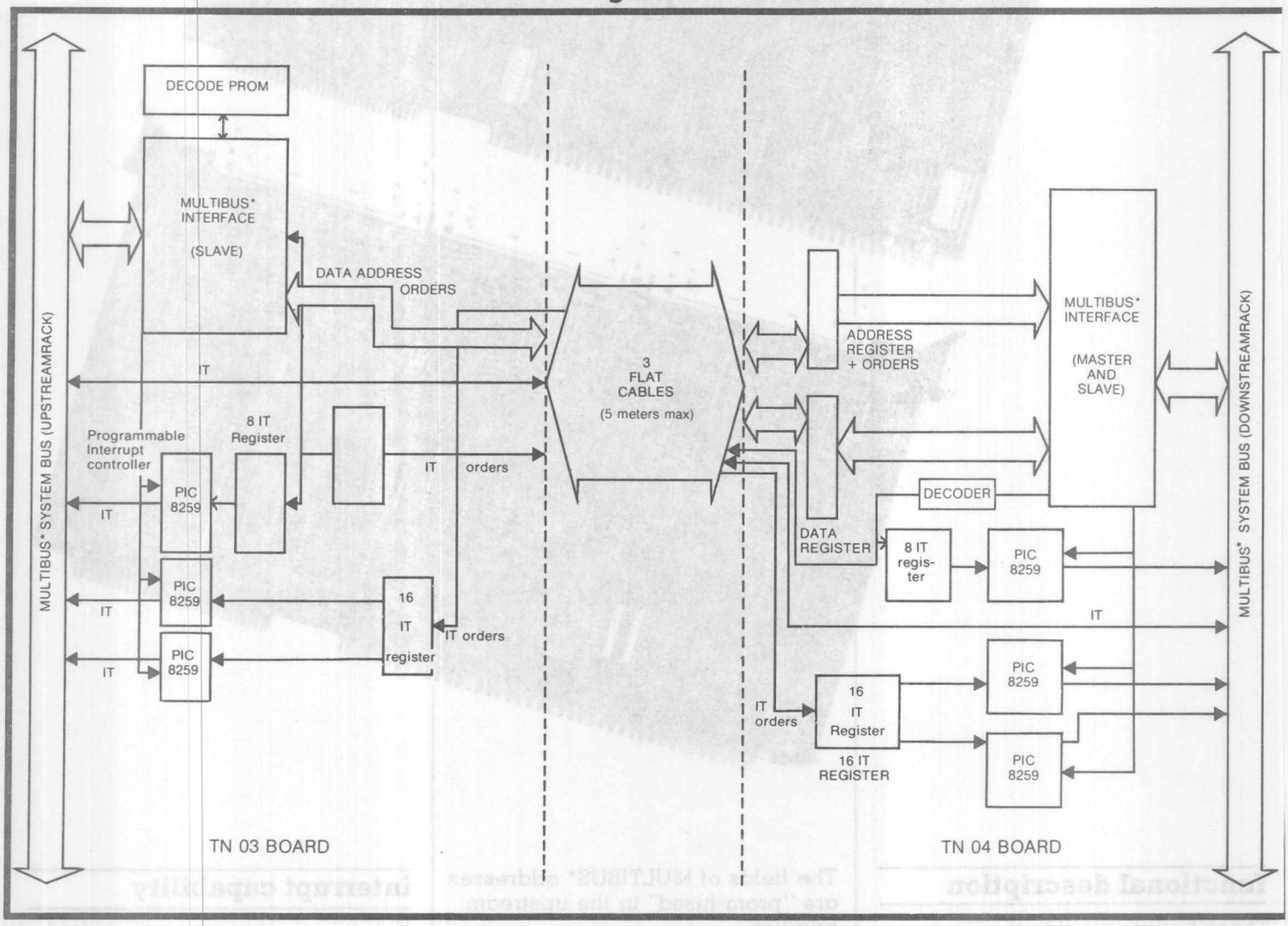
Downstream bus addresses may be obtained by transcoding upstream bus addresses so that one master board may command several identical slave systems.

Maximum addressing possibilities:
1 Mbytes in memory addressing
64 Kbytes in I/O addressing.

multimaster capabilities

TN 04 board provides full Multibus* arbitration control logic which allows up to three bus masters with serial priority, and up to 12 masters with an internal priority network.

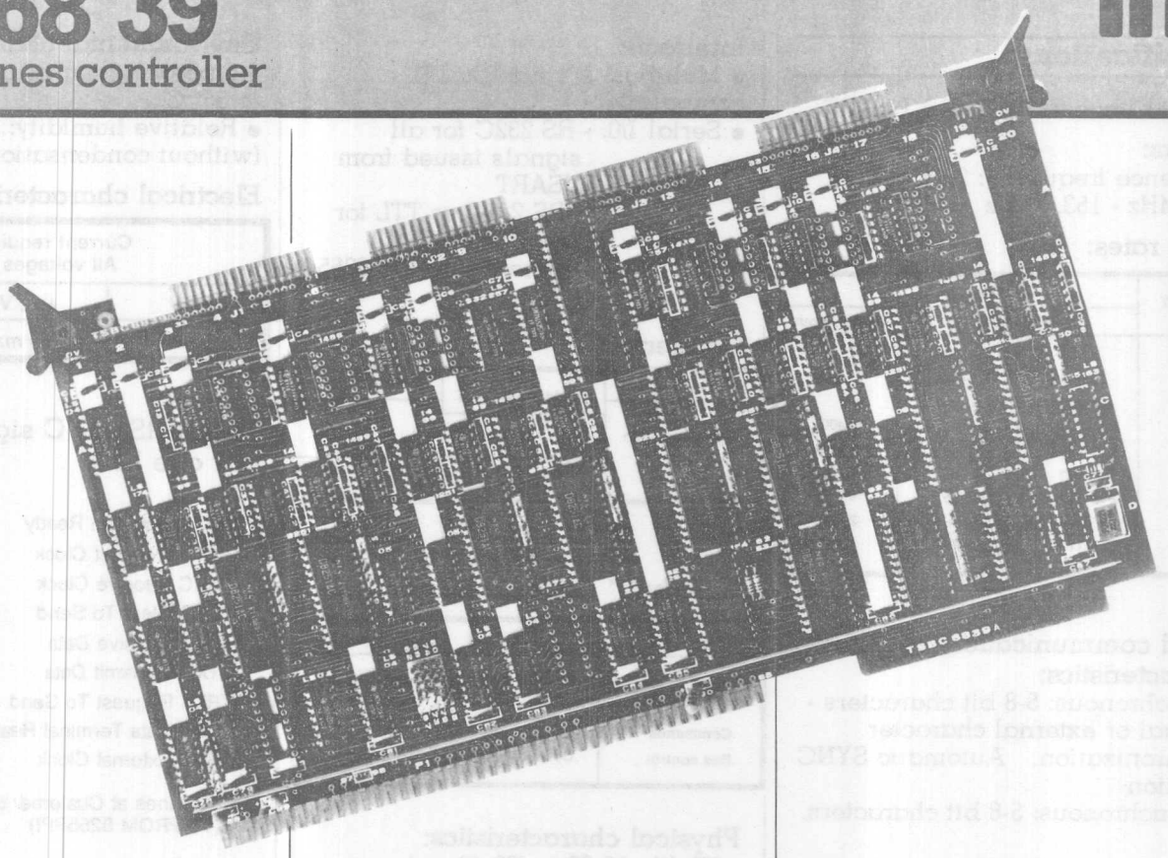
block diagram TN 03 / TN 04



TN 68 39

8 serial lines controller

TITN



- Provides 8 asynchronous or synchronous serial lines.
- Supports RS 232C interface.
- Three 8253 programmable timers with three programmable 16 bit BCD or binary timers/event counters each.
- 16 sources of interrupts (2 by serial line) strappable on any of 8 Multibus* interrupts.
- Baud rate generator up to 38.400 bauds synchronous and 9.600 bauds asynchronous programmable by software.
- A complete family of single board computers, memory, peripheral controllers, packaging and software is available.

The board is a 6.75 x 12.00 inches printed circuit board.

functional description

The TN 68 39 board is a Multibus* compatible slave board. It provides 8 programmable communication interfaces using the Intel 8251A universal synchronous/asynchronous receiver/transmitter (USART). Each counter output of a programmable timer is dedicated to a USART as a clock. The USART may use either this clock or an external clock.

The receive or transmit clocks may be common or independent. If they are internal and independent, they must be identical for two USART's of sequential rank. The selection of the receiver and transmitter clocks is done by straps in a matrix.

A parallel peripheral interface (8255) adds three lines at each RS 232C interface. These lines may be used as input or output lines at customer discretion. The choice of the direction must be the same for all the serial lines.

Sixteen sources of interrupts, one for transmission (TX ready), one for reception (RX ready) are stored in two eight bit registers: TX ready register and RX ready register. These two registers may be read through the Multibus.* An interrupt matrix provides the multiplexing of the 16 outputs of the two registers on any number of the 8 Multibus* interrupt lines.

specifications

8 serial lines using 8251A USART

Timers:

Reference frequency: 2.46 MHz - 1.23 MHz - 153.6 KHz

Baud rates:

Frequency KHz	Baud rate		
	Asynchronous		Synchronous
	: 16	: 64	
153.6	9.600	2.400	
76.8	4.800	1.200	
38.4	2.400	600	38.400
19.2	1.200	300	19.200
9.6	600	150	1.600
4.8	300	75	4.800
2.4	150	—	2.400
1.76	110	—	1.760

Serial communication characteristics:

- Synchronous: 5-8 bit characters - internal or external character synchronization. Automatic SYNC insertion
- Asynchronous: 5-8 bit characters.

Interface:

- Multibus*: All signals TTL compatible
- Serial I/0: - RS 232C for all signals issued from USART
  - RS 232C or TTL for the three signals connected to the 8255 PPI

Connectors

Interface	Double sided pins	Centers
Multibus* system	P1 = 86 pins	0.156
Serial I/0	34 pins (for two serial lines)	0.1

Multibus* drivers

Function	Characteristics	Sink current (ma)
Data	Tristate	32
Address	Tristate	32
Commands	Tristate	32
Bus control	Open collector	20

Physical characteristics:

- Width: 12.00 in (30.48 cm)
- Height: 6.75 in (17.15 cm)
- Depth: 0.70 in (1.78 cm)

Environmental characteristics:

- Operating temperature: 0° C to 55° C
- Relative humidity: to 90% (without condensation)

Electrical characteristics

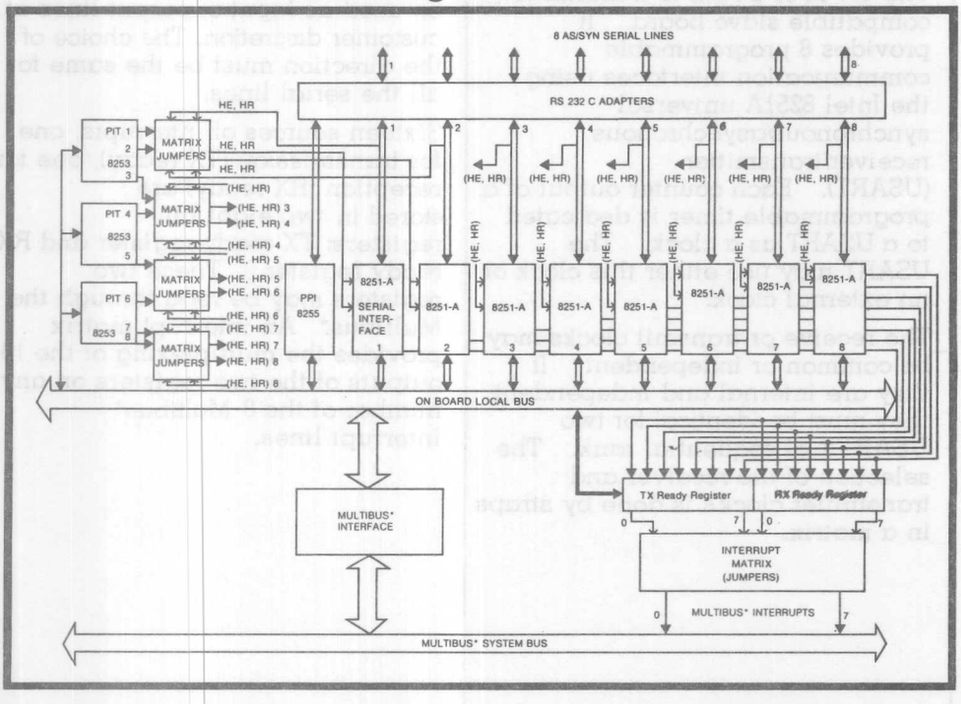
Current requirement All voltages ± 5%		
+5 V	+12 V	-12 V
3,5 A	250 ma	250 ma

Serial RS 232 C signals (for one line)

DSR Data Set Ready
TXC Transmit Clock
RXC Receive Clock
CTS Clear To Send
RXD Receive Data
TXD Transmit Data
RTS Request To Send
DTR Data Terminal Ready
EXC External Clock

+ 3 lines at Customer discretion (TO/FROM 8255PPI)

block diagram TN 68 39



IN 68/18

CLOCK, CALENDAR AND RAM WITH ON BOARD BATTERY BACK-UP

- 8 Kbytes of RAM
- 24 hour clock with 0,1 s resolution
- CALENDAR function with month, day of week, day of (year/month)
- One programmable timer
- On board battery back up with 10 days capacity.
- Parallel priority management for up to 12 Master boards
- Multibus* interface for slave operation
- A complete family of single board computers, memory, peripheral controllers, packaging and software is available.

FUNCTIONAL DESCRIPTION

The TN 68/18 provides several independent facilities, available to master boards of Multibus* systems.

RAM CAPABILITY :

The TN 68/18 board includes 8 Kbytes of slave memory, available for Multibus* master boards via the system bus. The memory is addressed as one segment of 8 Kbytes in a field of 1 Megabyte. The address of the field is defined by straps. In case of power supply failure, transmitted via the Multibus*, the power is supplied by a battery, with an guaranteed capacity of 10 days. An associated logic manages all the Multibus* signals dedicated to the power failure : ACLO, PFIN, PFSN, MPRO, INIT.

CLOCK AND CALENDAR :

The TN 68/18 provides the calendar information in the following format of BCD characters :

Years / Tens of months / Units of months / Tens of days of month / Units of days / Tens of hours / Units of hours / Tens of Minutes / Units of minutes / Tens of seconds / Units of seconds / Tenths of seconds.

These characters may be read as the least significant four bits of a byte through Multibus*. So, the day of the week may be read. The updating may be done by loading of the minutes, hours, days and years.

* Trademark from Intel

The start and stop of the calendar are under program control. The synchronization of the reading is done by a test on the value of the character read. If it is 1111, this indicates that the counters are changing and that the reading must be resumed.

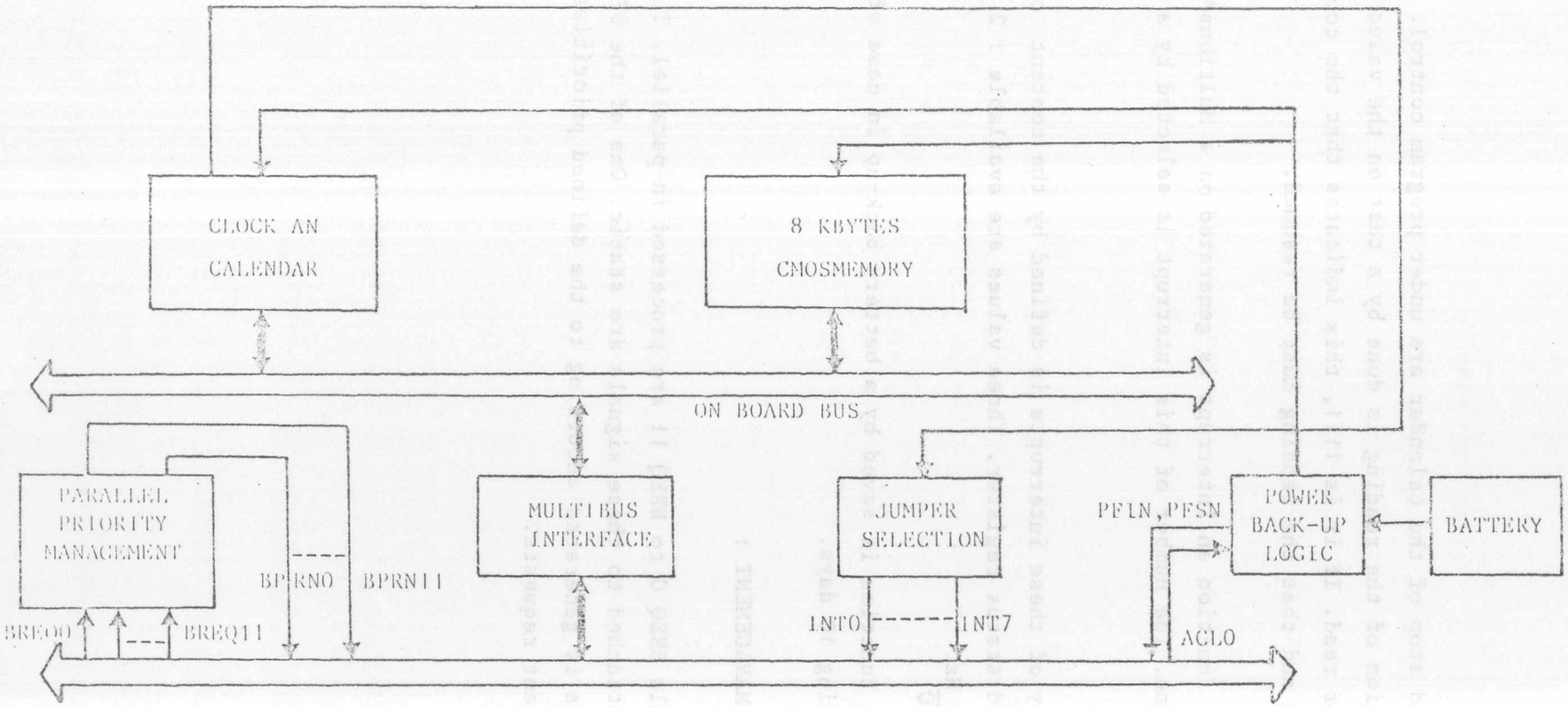
In the clock function an interrupt is generated on a Multibus* interrupt line. The number of this interrupt is selected by a jumper.

The frequency of these interrupts is defined by the content of the interrupt and status register. Three values are available : 2 Hz, 0.2 Hz and $\frac{1}{60}$ Hz.

The calendar function is saved by a battery back-up in case of power failure, during 10 days.

PARALLEL PRIORITY MANAGEMENT :

The 12 signals BREQ 0 to BREQ 11 are processed in parallel. The priorities attached to these signals are static. One of the BPN 0 to 11 signals is generated according to the defined priorities and the current requests.


CLOCK AN
CALENDAR
8 KBYTES
CMOSMEMORY
ON BOARD BUS
PARALLEL
PRIORITY
MANAGEMENT
MULTIBUS
INTERFACE
JUMPER
SELECTION
POWER
BACK-UP
LOGIC
BATTERY
PFIN, PFSN
BPRN0
BPRN11
BREQ00
BREQ11
INT0
INT7
ACL0

SPECIFICATIONS

Memory : - 8 Kbytes with a modularity of 1 Kbyte.
- Base address defined by straps in 1 Mbyte field.
- Jumpers to define the on board modules.
- Maximum access time : 740 ns
- Selection of use of advanced acknowledge by strap.

Clock and calendar :

- 16 sequential address registers are dedicated to the clock and calendar functions. These addresses are referenced to a base address defined by straps.
- Format of registers : - 4 LSB are used
  - MSB copies the PFSN status
- The tables of Annex I describes the function an relative address of the registers.

Logic for back-up on power failure :

- Input : AC LOW
- Outputs : Power fail interrupt
  Power fail sense
  Initialization
  Memory protect

Battery :

- Capacity : 100 mAh
- Voltage : 3,6 volts
- Maximum loading time : 20 hours

Parallel priority management : up to 12 static signals.

Connectors :

	Double sided pins	Centers (inches)
P1 =	86	0,156
P2 =	50	0,1

Multibus interfaces : TTL compatible

Multibus drivers :

Function	Characteristics	Sink current (ma)
Data	Tri state	32
Address	Tri state	32
Commands	Tri state	32
Bus control	Open collector	20

Physical characteristics :

Width : 12.00 in (30,48 cm)
Height : 6.75 in (17,15 cm)
Depth : 0.70 in (1,78 cm)

Environment characteristics :

Operating temperature : 0° C to 55° C
Relative humidity : to 90 % (without condensation)

Electrical characteristics :

Voltage : 5 volts $\pm$ 0,5 V
Current : 2,2 A max.

Table 1 Address Decoding For Internal Registers

SELECTED COUNTER	ADDRESS BITS				MODE
	AD 3	AD 2	AD 1	AD 0	
0 Test only	0	0	0	0	Write only
1 Tenths of secs	0	0	0	1	Read only
2 Units of secs	0	0	1	0	Read only
3 Tens of secs	0	0	1	1	Read only
4 Units of mins	0	1	0	0	Read or Write
5 Tens of mins	0	1	0	1	Read or Write
6 Units of hours	0	1	1	0	Read or Write
7 Tens of hours	0	1	1	1	Read or Write
8 Units of days	1	0	0	0	Read or Write
9 Tens of days	1	0	0	1	Read or Write
10 Day of week	1	0	1	0	Read or Write
11 Units of months	1	0	1	1	Read or Write
12 Tens of months	1	1	0	0	Read or Write
13 Years	1	1	0	1	Write only
14 Stop/Start	1	1	1	0	Write only
15 Interrupt & Status	1	1	1	1	Read or Write

Table 2A Interrupt Selection Data

MODE: ADDRESS 15, WRITE MODE				
Function	DB3	DB2	DB1	DB0
No Interrupt	x	0	0	0
Int. at 60 sec intervals*	0/1	1	0	0
Int. at 5.0 sec intervals*	0/1	0	1	0
Int. at 0.5 sec intervals*	0/1	0	0	1
* ±16.6ms				

DB3 = 0, single interrupt DB3 = 1, repeated interrupt

Table 2B Interrupt Read Back (Status)

MODE ADDRESS 15, READ MODE				
Interrupt Status	DB3	DB2	DB1	DB0
Reset	0	0	0	0
60 sec signal	0/1	1	0	0
5.0 sec signal	0/1	0	1	0
0.5 sec signal	0/1	0	0	1

DB3 = 0, no interrupt DB3 = 1, interrupt from timer

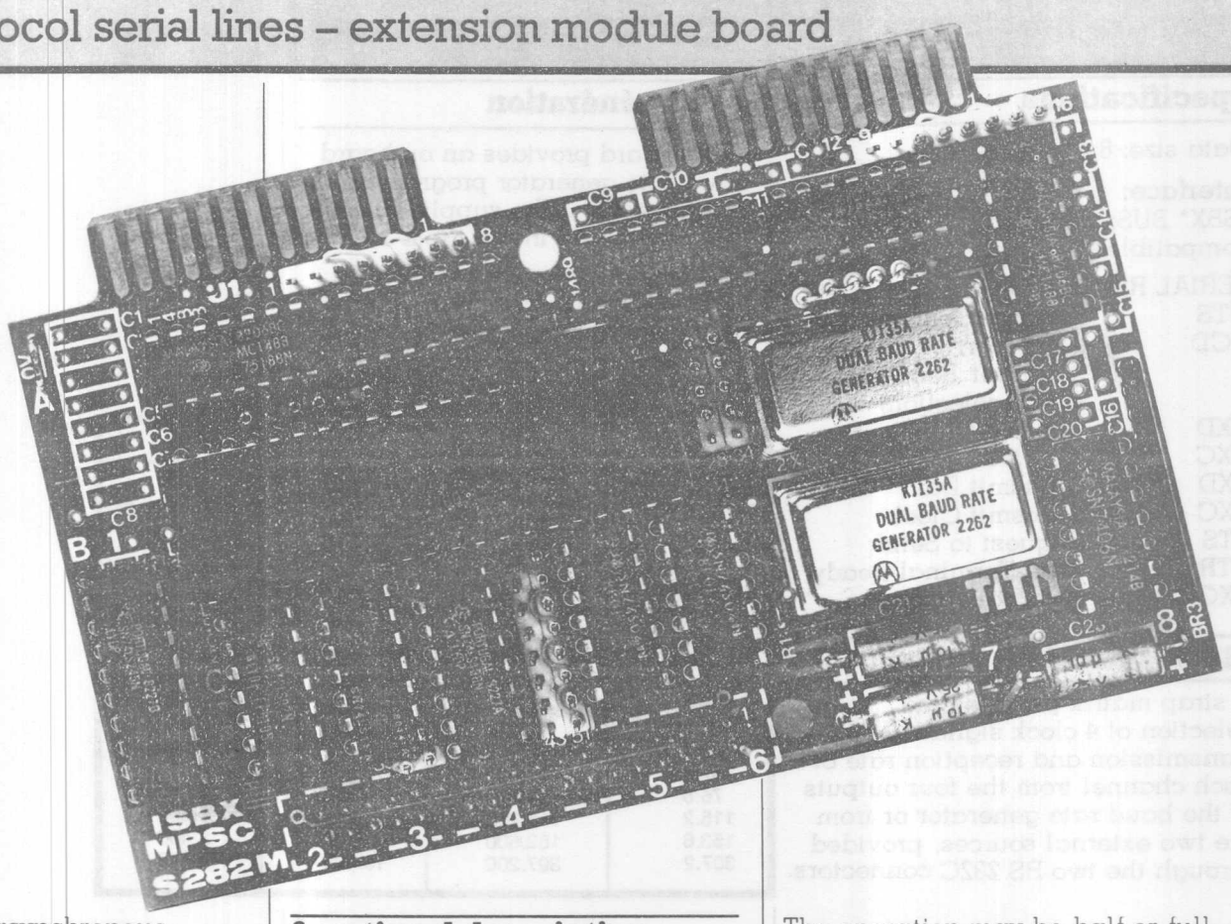
Table 3 Years Status Register

MODE: ADDRESS 13, WRITE MODE	DB3	DB2	DB1	DB0
Leap year	1	0	0	0
Leap year + 1	0	1	0	0
Leap year + 2	0	0	1	0
Leap year + 3	0	0	0	1

TNBX 05

2 multiprotocol serial lines – extension module board

TITN



- Provides two asynchronous, synchronous, HDLC/SDLC half/full duplex serial lines for iSBX* bus compatible microcomputers.
- 4 independent on-board baud rate generators, programmable by software.
- Supports RS 232C interface.

The TNBX 05 provides incremental on board I/O expansion support for asynchronous, synchronous and ISO/CCITT's HDLC or IBM's SDLC communication. Plugging directly into any iSBX* bus compatible host board, the TNBX 05 module provides two R232C communication channels with 4 on-board software selectable baud rates, up to 307.200 Kbauds for full-duplex synchronous operations.

Once installed on the mother-board iSBX* connector the TNBX 05 daughter board makes the mother board thicker.
Complying with INTEL specifications, the resulting assembly requires two consecutive slots.

functional description

The TNBX 05 uses the MPSC 8274 programmable multiprotocol controller. This controller links the parallel TNBX 05 to the two serial channels.

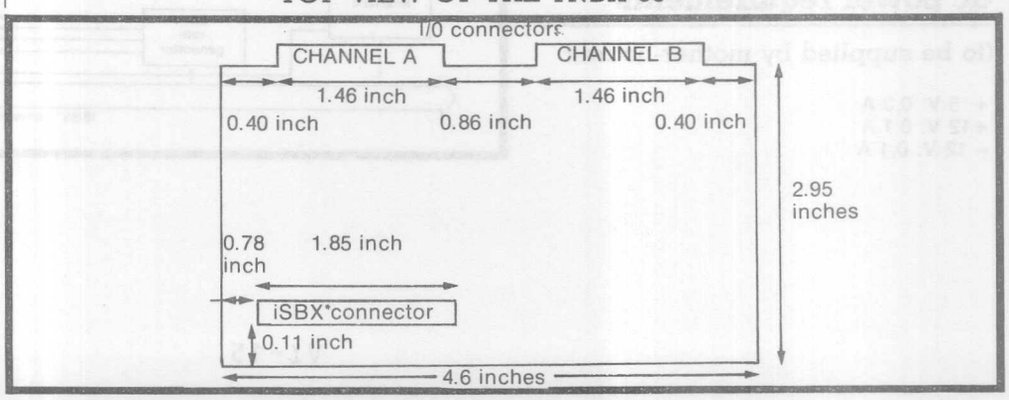
During a transmit sequence, the TNBX 05 accepts data and commands from the iSBX*bus interface, translates and formats the data in asynchronous, synchronous or HDLC/SDLC protocol formats, provides the RS 232C interface control signals and sends the data to the serial channels.

The operation may be half or full duplex. 4 independent on-board baud rates are programmable by solftware.

The frequency inputs of the controllers may be selected through a strap matrix from these 4 baud rate generators or from 2 external sources provided through the RS 232C interface external clock.

The strap matrix allows selection of the DCE or DTE mode and the crossing or looping of TXD and RXD and of RTS, CTS, DCD signals.

TOP VIEW OF THE TNBX 05



Trademark from Intel Corporation.

specifications

Data size: 8 bits

Interface:
iSBX* BUS: All signal TTL compatible

SERIAL RS 232 C SIGNALS:

CTS	Clear to Send
DCD	Data Carrier Detect or Data Set Ready (strap selection)
RXD	Receive Data
RXC	Receive Clock
TXD	Transmit Data
TXC	Transmit Clock
RTS	Request to Send
DTR	Data Terminal Ready
EXC	External Clock

clock selection

A strap matrix provides the selection of 4 clock signals for the transmission and reception rate of each channel from the four outputs of the baud rate generator or from the two external sources, provided through the two RS 232C connectors.

Characteristics of the synchronous mode:
— 5 to 8 bit characters—Automatic SYN insertion—Internal or external character synchronization.

Characteristics of the asynchronous mode:
— 5 to 8 bit character mode
— 1, 1 1/2 or 2 stop bits
— break character generation
— false start bit detection

Characteristics of HDLC and SDLC bit synchronous mode:
— These bit synchronous modes are fully compatible with the CCITT or IBM specifications.

dc power requirements

(to be supplied by mother-board):

+ 5 V: 0,3 A
+12 V: 0,1 A
– 12 V: 0,1 A

baud génération

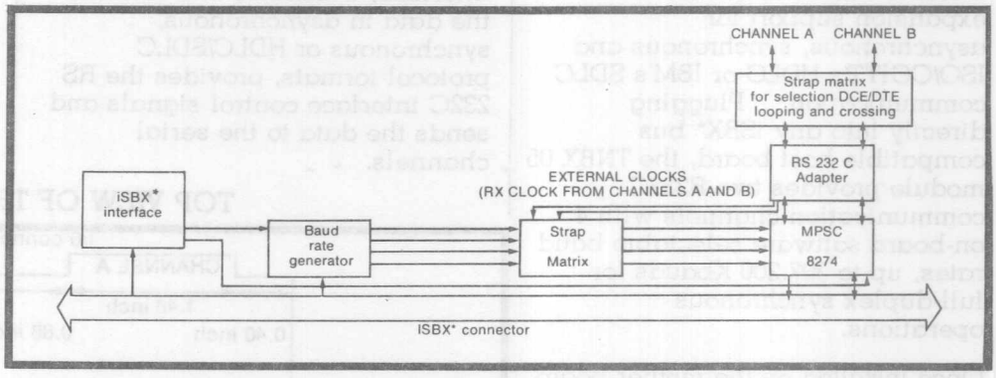
• The board provides an on-board baud rate generator programmable by solftware. The supplied baud rates are given in the table 1.

• Using an external clock, MPSC 8274 may run at up to 880 K bauds in synchronous mode.

On-board baud rate. (table 1)

Selected baud rate K baud / Mode	Synchronous or HDLC BAUDS	Asynchronous	
		: 16	: 64
0.8	800	50	
1.2	1.200	75	—
1.76	1.760	110	—
2.152	2.152	134.5	—
2.4	2.400	150	—
4.8	4.800	300	75
9.6	9.600	600	150
19.2	19.200	1.200	300
28.8	28.800	1.800	450
32	32.000	2.000	500
38.4	38.400	2.400	600
57.6	57.600	3.600	900
76.8	76.800	4.800	1 200
115.2	115.200	7.200	1.800
153.6	153.600	9.600	2.400
307.2	307.200	19.200	4.800

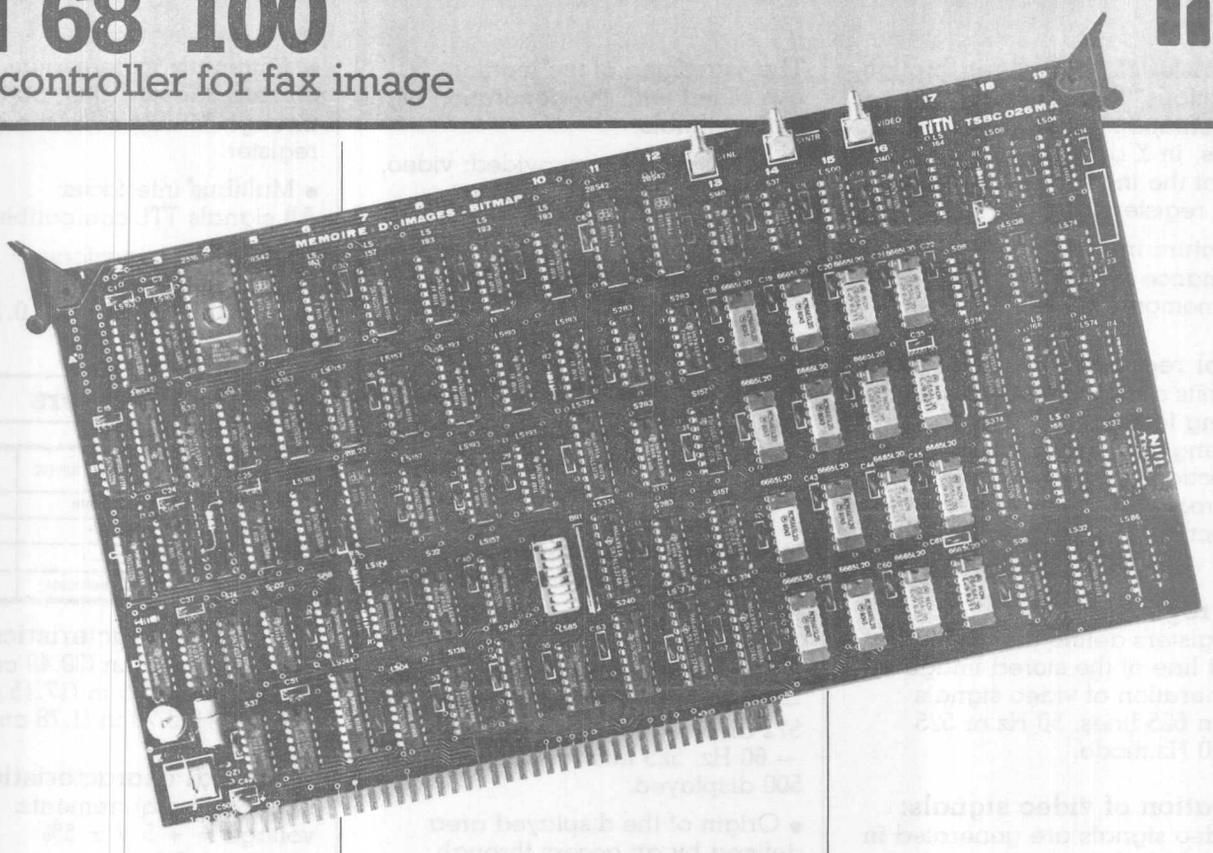
Block diagram of the TNBX 05



TN 68 100

video controller for fax image

TITN



- Addressable image: 864 X 1144
- Image displayed: 864 X 572 for 50 Hz
- Vertical scroll by use of a 11 bits scroll register
- Multibus* interface for slave operation
- A complete family of single board computers, memory, peripheral controllers, packaging and software is available
- Half horizontal A4 FAX definition
- Vertical A4 FAX definition

functional description

The TN 68 100 board is dedicated to the Multibus* slave mode. It includes a binary RAM memory, picturing an image of horizontal resolution equal to half a FAX horizontal resolution (i.e $\frac{1728}{2}=864$) and of vertical definition equal to the vertical FAX, i.e. 1144.

This memory may be accessed through the Multibus*: in writing by byte or bit and in reading by byte.

The video signals may be selected for a 625 lines monitor type, interlaced at 50 Hz or 525 lines type, interlaced at 60 Hz. The memory is accessed through the Multibus* as a set of 8 registers of sequential addresses relative to a base, defined by switches on the board.

addressing of the memory

Two registers, X and Y, provide the addressing of the memory through the Multibus*, with a resolution of 1144 by 864. The X register (10 bits) addresses a bit in a line of 864 bits. The Y register (11 bits) is loaded with the number of the addressed line, with a limit of 1144. The memory is organized in 64K words of 16 bits. So, a line of the image fills in 54 words of the memory.

In case of bit addressing, the address of this bit in the addressed byte is given by the three LSB of X.

The mode of addressing is defined by a bit of the control register.

data registers

There are two data registers:

Write data register:

This byte register is loaded through Multibus* with the data to be written, at the byte address defined by the X and Y registers. In byte mode, the data is the byte to be written. In bit mode, only the bit at the position in the write data register defined by the three LSB of X register is written in the memory.

As soon as this register is loaded through the Multibus*, the writing in memory is executed. There is also an incrementation of the byte address. This incrementation may be on the X or on the Y register, depending on the status of the bit of the control register which selects the incrementation mode.

So it is easy to fill the memory, according to the X or Y direction.

Read data register:

This byte register is loaded with the content of the byte addressed by the registers X and Y, as soon as the Y register is loaded through the Multibus*.

* Trademark from Intel Corporation

The transfer of this register through the Multibus* causes the incrementation of the current address, in X or Y, in function of the status of the increment bit of the control register.

This feature improves the performance in transfering an area of the memory.

Control register:
Three bits control independently the following features:
— Writing of bytes or bit
— Selection of incrementation in X or Y direction
— Selection of normal or inverted video.

Scroll registers:
Two registers define the number of the first line of the stored image for the generation of video signals either in 625 lines, 50 Hz or 525 lines, 60 Hz mode.

Generation of video signals:
The video signals are generated in parallel with the accesses to the memory through the Multibus*. At the maximum, half of the availability of the memory is dedicated to the generation of the video signal.

The refreshing of the memory is combined with the generation of the video signals.

Three signals are provided: video, line sync and frame sync.

specifications

- Size of the memory: 128 Kbytes
- Addressable image: 864 x 1144
- Type of access through Multibus*
  — Writing access: byte or bit
  — Reading access: byte
- Programmable controls:
  — normal or inverted video
  — selection of incrementation in X or Y
  — writing access by byte or bit.
- Type of video generated:
  — 50 Hz, 625 interlaced lines with 572 displayed
  — 60 Hz, 525 interlaced lines with 500 displayed.
- Origin of the displayed area defined by an access through Multibus* register.
- Automatic incrementation of the current address after transfer through Multibus* of the read register.
- Automatic incrementation of the current address after loading through Multibus* of the write register.
- Multibus* interfaces:
  All signals TTL compatible.
- Multibus* connectors:
  86 double sided pins
  distance between pins: 0,156 inch.

Multibus* drivers

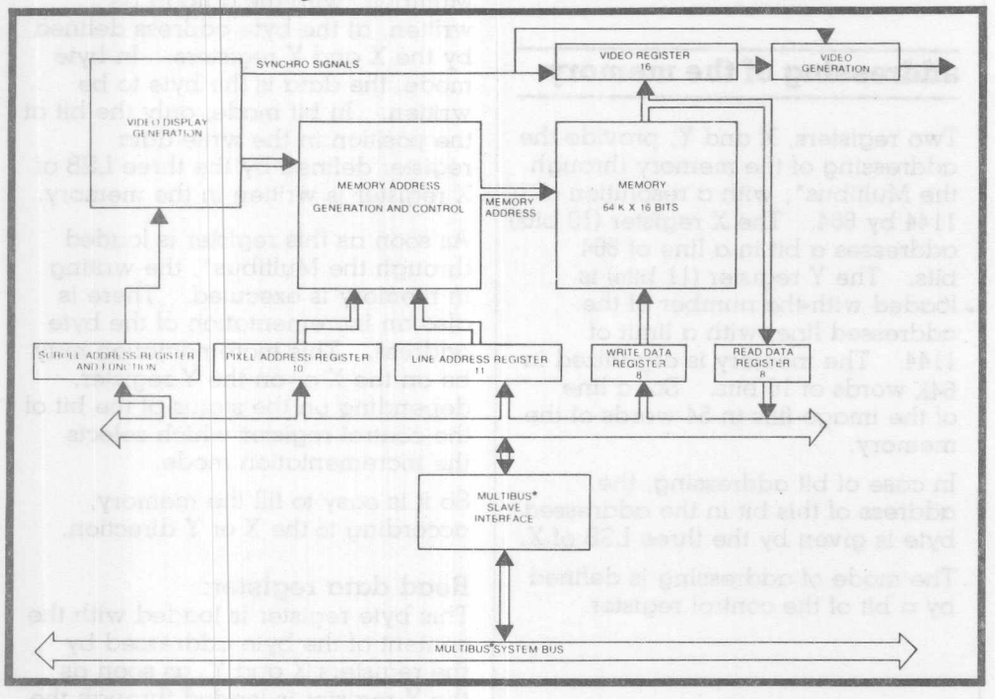
FUNCTION	CHARACTERISTIC	SINK CURRENT
Data	Tristate	32
Address	—	—
Commands	—	—
Bus control	Open collector	20

Physical characteristics:
- WIDTH: 12.00 in (30.48 cm)
— HEIGHT: 6.75 in (17.15 cm)
— DEPTH: 0.70 in (1.78 cm)

Electrical characteristics:
DC power requirements:
voltage = + 5 V ± 5%
current = 3,5 A

Environmental characteristics:
— Operating temperature: 0° C to 55° C
— Relative humidity: to 90% (without condensation).

block diagram of TN 68 100 video controller



TN68110

CONTROLEUR VIDEO COULEUR COMPATIBLE MULTIBUS*

- Résolution : 256 x 256

- 50 Hz non entrelacé

- Vraie couleur
  - . 6 bits/pixel pour chaque couleur primaire (R, V, B)
  - . 8 types de correction de couleur programmable

- Accès simultané aux 3 plans en vraie couleur possible

- Plan supplémentaire Fausse couleur
  - . Forçage ou inhibition sélective sur chaque composante

- Effacement matériel
  - . Plan supplémentaire
  - . Sélectivement ou simultanément sur les plans couleur

- Auto incrément de l'adresse possible
  - . Suivant l'axe vertical ou horizontal

- Affichage en rouleau à 1 ligne près

- Arrêt vidéo programmable

- Carte compatible MULTIBUS*

DESCRIPTION FONCTIONNELLE

La TN68110, carte au format SBC d'INTEL, fonctionne sur le MULTIBUS* en mode esclave .

Elle dispose d'une mémoire 64K x 19 bits, soit 6 bits par couleur (64 niveaux) et un bit permettant l'élaboration du plan graphique .

* MULTIBUS marque déposée d'INTEL CORPORATION.

T.055/08.83

Les informations contenues dans ce document sont susceptibles de changer.

CARACTERISTIQUES COULEUR

Les 3 composantes couleur de 6 bits chacune, pour un pixel situé à l'adresse définie par X et Y, sont accessibles de 2 manières différentes :

Accès direct : L'accès est direct pour la composante désirée; c'est à dire que chaque composante est accédée par une adresse, (en écriture et lecture).

Accès indirect : L'accès indirect se fait par l'intermédiaire d'un registre d'indirection de 3 bits donnant pour chaque bit la composante à accéder.

En écriture l'accès peut être multiple; c'est à dire que l'on peut écrire une donnée simultanément dans plusieurs plans.

En lecture l'accès est unique et se fait prioritairement dans le sens RVB.

Plan supplémentaire

Le plan supplémentaire, composé de 1 bit par pixel est accèdé directement au moyen d'un registre propre en lecture et écriture.

On peut également le lire avec chaque composante couleur par une configuration prévue à cet effet sur la carte.

CARACTERISTIQUES VISUELLES

Un registre pointeur de page est accessible par le bus. Il permet d'effectuer la fonction de Rolling sur l'image.

ACCES MEMOIRE

Les accès mémoire sont partagés entre la vidéo et le MULTIBUS*

- accès MULTIBUS* toutes les 3,8 µs en mode normal
- accès MULTIBUS* toutes les 400 ns en arrêt vidéo

Un jeu de 12 registres permet, à travers le MULTIBUS* d'accèder à la mémoire :

0 Registre indirection (E)
1 Registre d'accès au plan par indirection (E et L)
2 Registre plan rouge (E et L)
3 Registre plan vert (E et L)
4 Registre plan bleu (E et L)
5 Registre plan graphique (E et L)
6 Coordonnée X (E)
7 Coordonnée Y (E)
8 Flag effacement (L)
9 Registre fonction (E)
A Registre attribut du plan graphique (E)
B Registre de Scroll (E)
C Registre de sélection de correction

NOTE : E = Accès en écriture seulement
E et L = Accès en écriture et lecture
L = Accès en lecture seulement

SPECIFICATIONS

Interface MULTIBUS* :

- carte esclave
- bus de données 8 bits

Sorties vidéo :

- niveau de sortie ajustable, 1V à 5V, sur chaque composante
- synchro ligne plus trame TTL en logique positive ou négative

Caractéristiques physiques :

Largeur : 12 in. (30,48 cm)

Hauteur : 6,75 in. (17,15 cm)

Epaisseur : 0,70 in. (1,78 cm)

Caractéristiques électriques :

Consommation sur le 5V : 3 A max.

Consommation sur le + 12V : 170 mA.

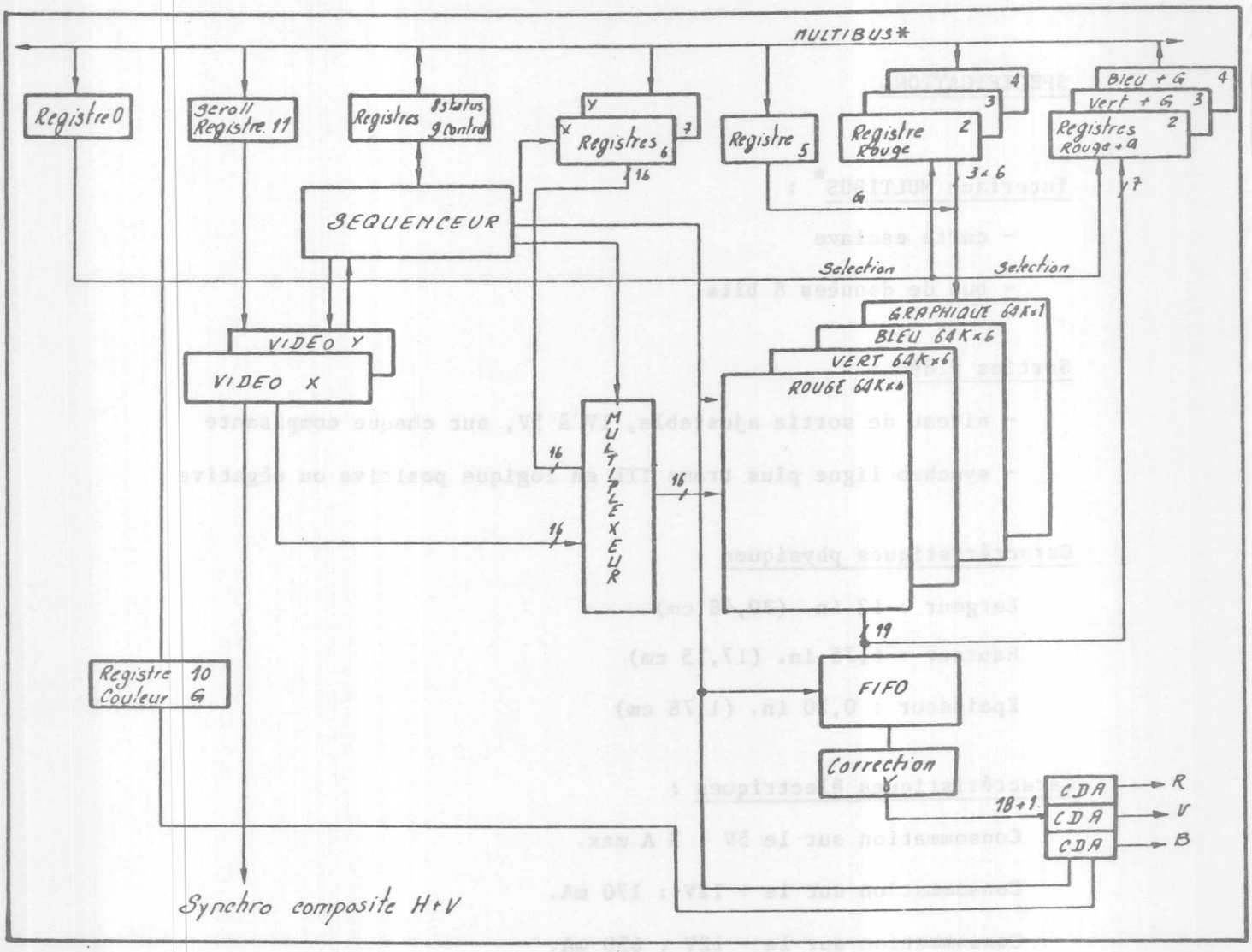
Consommation sur le - 12V : 650 mA.

Caractéristiques du moniteur vidéo :

. Fréquence horizontale : 15625 Hz
Suppression ligne minimum : 12 µs

. Fréquence verticale : 50 Hz
Suppression composite horizontale - verticale niveau TTL - logique positive ou négative

. 3 entrées couleur RVB 1 à 4 V.



TN68110 - SCHEMA SYNOPTIQUE

THOMSON-TITN

SIÈGE SOCIAL 1 à 5, rue Gustave Eiffel - 91420 **MORANGIS** - Tél. (6) 909.34.44 - Télex 691.163

55, rue Edgar Quinet - 92242 **MALAKOFF** - Tél. (1) 657.11.10

rue Denis Papin - Z.I. la Vigne aux Loups - 91380 **CHILLY-MAZARIN** Dpt. Inf. Temps Réel - Tél. (6) 448.75.00

- 7 -

CARTES COMPATIBLES PLESSEY

°_°_°

LA SOCIETE PLESSEY EST DEPUIS LONGTEMPS UNE SECONDE SOURCE DES CARTES D'EXTENSION MEMOIRE AU FORMAT MULTIBUS.

EN UTILISANT LES COMPOSANTS LES PLUS PERFORMANTS DU MARCHE ET EN RESPECTANT LES NORMES PRECONISEES PAR iNTEL, RELATIVES AUX BUS MULTIBUS ET LBX, DE NOUVELLES CARTES SONT PROPOSEES :

- PSM 512 DA : . MEMOIRE 512 KO + ECC
  . ACCES MULTIBUS ET LBX

  C'EST UNE SECONDE SOURCE DE LA SBC 012 CX D'iNTEL

- PSM 6663 : . MEMOIRE CMOS RAM/EPROM
  . CAPACITE < 256 KO EN RAM
  < 1 MO EN EPROM

- PSM 2DA : . MEMOIRE DRAM 2 MO + ECC
  . ACCES MULTIBUS ET LBX

CES TROIS CARTES SONT A RETENIR DANS LES APPLICATIONS ORGANISEES AUTOUR DU iAPX 186 OU iAPX 286.

DANS UN AUTRE REGISTRE, PLESSEY PROPOSE UN EQUIPEMENT QUI PRESENTE MECANIQUEMENT LES CARACTERISTIQUES D'UN FLOPPY 5"1/4 SLIMLINE ET ELECTRIQUEMENT UN DISPOSITIF DE MEMOIRE A BULLE D'UNE CAPACITE DE 1/2 MO. IL S'AGIT DU PBU 85E.

L'INTERFACE ENTRE LE PBU 85E ET LE CALCULATEUR S'EFFECTUE AU TRAVERS D'UNE LIAISON RS 423 (RS 232C) OU RS 422. CES PROTOCOLES SIMPLIFIES PERMETTENT D'ATTEINDRE DES VITESSES DE TRANSFERT ALLANT JUSQU'A 38,4 K BAUDS.

GRACE A CET EQUIPEMENT IL DEVIENT AISE ET PEU COUTEUX DE REMPLACER UN DISQUE SOUPLE PAR UNE MEMOIRE A BULLE DANS DES ENVIRONNEMENTS SEVERES (CHOCS, VIBRATIONS, POLLUTION) OU LORSQUE LA FIABILITE EST UN PARAMETRE DETERMINANT.

Advanced information

PSM 512DA

Dual port Multibus (IEEE 796) and iLBX compatible Memory

512 k x 8 with EDC

Features

Reliability	Over 10 years calculated MTBF Uses high quality, multi-sourced 64 kDRAMs
Interfaces	Full compatibility with IEEE 796 and iLBX (both optimised and non optimised).
Capacity	512, 256 and 128 kbyte options
	24 bit addressing on both Multibus and iLBX local bus
Flexibility	
	Internal or external refresh control
	Compatible with all 8 and 16 bit processors
	Battery back-up option
Integrity	Detection of double bit errors Detection and correction of single bit errors
	Two I/O ports for diagnostics
Performance	250 ns access time (in non correcting mode)

Capacity	512 kbyte with 384, 256 and 128 kbyte options
Word length	8 or 16 bits
Cycle time	400 ns read or write
Access time	300 ns in correcting mode 250 ns in non-correcting mode
Operating modes	Read Write Refresh I/O Read I/O Write
Voltage requirement	5v ± 5% dc
Power requirement	Operating 5A Maximum Standby 2.5A (battery back-up)
Interface	Full compatibility with IEEE 796 (Intel Multibus) and iLBX (both optimised and non optimised). 24 bit addressing on both buses
Size	One IEEE 796 card slot
Weight	1 Kg

Operating environment

Temperature	0 to +55°C
Relative humidity	Up to 95% without condensation
Thermal shock	± 1°C/minute
Altitude	-300 to +3,000 metres
Vibration	5 to 100 Hz with 0.5g acceleration (half sine) when mounted in a suitable racking system

Storage environment

Temperature	-55 to +85°C
Relative humidity	Up to 95% without condensation
Thermal shock	± 10°C/minute
Altitude	-300 to +10,000 metres
Vibration	0 to 500 Hz with 2g acceleration
Mechanical shock	20g for 6 milliseconds (half sine)

Advanced information

PSM 6663

Multibus (IEEE-796)

CMOS RAM/EPROM

Real Time Clock option

Features

Reliability	Over 50,000 hours calculated MTBF
Interfaces	Full compatibility with IEEE-796
Capacity	CMOS RAM: Up to 256 kbytes EPROM: Any device up to 256 kbit Maximum capacity 1Mbyte Depopulated versions available RAM/EPROM mix
Flexibility	Battery back-up option provides up to 1 year data retention using Lithium cell
Performance	200ns access time (RAM) * 300ns access time (EPROM) typical
Address field	24 bit addressing

Specification

Capacity	Up to 64 kbyte of CMOS RAM - 16k devices Up to 256 kbyte of CMOS RAM - 64k devices Up to 1 Mbyte EPROM RAM/EPROM mix (options available on request) 24 bit addressing
Word length	8 or 16 bits
Cycle time	200ns read or write (RAM) 300ns read (EPROM) typical
Access time	200ns(RAM) 300ns (EPROM) typical
Operating modes	Read Write I/O Read I/O Write (R.T.C. clock option)
Voltage requirement	5v ± 5% dc
Power requirement	Operating +5V 1.3A Standby +5V 1.1A Power down Battery 400mA at 3V
Interface	TTL
Size	IEEE-796 card slot
Weight	1 Kg

Operating environment

Temperature	0 to +40°C (with batteries fitted) 0 to +55°C (without batteries fitted)
Relative humidity	Up to 95% without condensation
Thermal shock	±1°C/minute
Altititude	-300 to +3,000 metres
Vibration	5 to 100Hz with 0.5g acceleration (half sine) when mounted in a suitable racking system

Storage environment

Temperature	0 to +40°C (with batteries fitted) -30 to +85°C (without batteries fitted)
Relative humidity	Up to 95% without condensation
Thermal shock	± 10°C/minute
Altititude	-300 to +10,000 metres
Vibration	0 to 500 Hz with 2g acceleration
Mechanical shock	20g for 6 milliseconds (half sine)

Advanced information

PSM 2DA

Dual port Multibus (IEEE 796) and iLBX compatible Memory

2 megabytes DRAM with EDC

Features

Reliability	Over 10 years calculated MTBF Uses high quality, multi-sourced 256 kDRAMs
Interfaces	Full compatibility with IEEE 796 and iLBX (both optimised and non optimised).
Capacity	2, 1.5, 1 and 0.5 Mbyte options
	24 bit addressing on both Multibus and iLBX local bus
Flexibility	
	Internal or external refresh control
	Compatible with all 8 and 16 bit processors
	Battery back-up option
Integrity	Detection of double bit errors Detection and correction of single bit errors
	Two I/O ports for diagnostics
Performance	250 ns access time (in non correcting mode)

Specification

Capacity	2 Mbyte with 1.5, 1 and 0.5 Mbyte options
Word length	8 or 16 bits
Cycle time	400 ns read or write
Access time	300 ns in correcting mode 250 ns in non-correcting mode
Operating modes	Read Write Refresh I/O Read I/O Write
Voltage requirement	5v ± 5% dc
Power requirement	Operating 5A Maximum Standby 2.5A (battery back-up)
Interface	Full compatibility with IEEE 796 (Intel Multibus) and iLBX (both optimised and non optimised). 24 bit addressing on both buses
Size	One IEEE 796 card slot
Weight	1 Kg

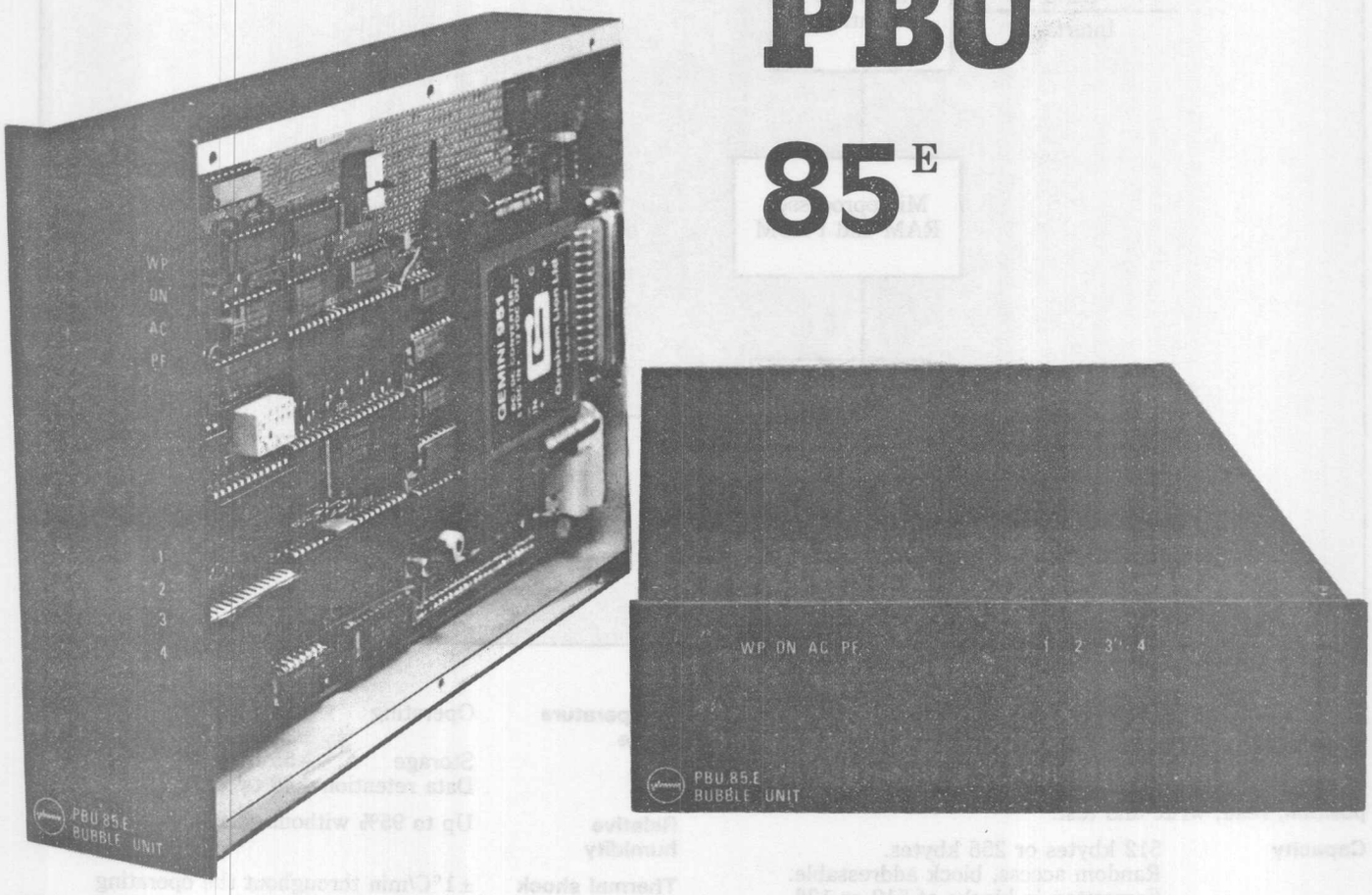
Operating environment

Temperature	0 to +55°C
Relative humidity	Up to 95% without condensation
Thermal shock	± 1°C/minute
Altitude	-300 to +3,000 metres
Vibration	5 to 100 Hz with 0.5g acceleration (half sine) when mounted in a suitable racking system

Storage environment

Temperature	-55 to +85°C
Relative humidity	Up to 95% without condensation
Thermal shock	± 10°C/minute
Altitude	-300 to +15,000 metres
Vibration	0 to 500 Hz with 2g acceleration
Mechanical shock	20g for 6 milliseconds (half sine)

PBU 85E



1/2 Megabyte Non-volatile Bubble Memory

More than ten years equipment life

Performance

PBU85E is an intelligent, high performance, bubble memory unit containing 512 kbytes of solid state memory in a standard 5¼" slimline floppy disc sized package.

PBU85E is used as a random access, block addressable memory, with a fast access time of 36 ms (average) for a block of 512 bytes.

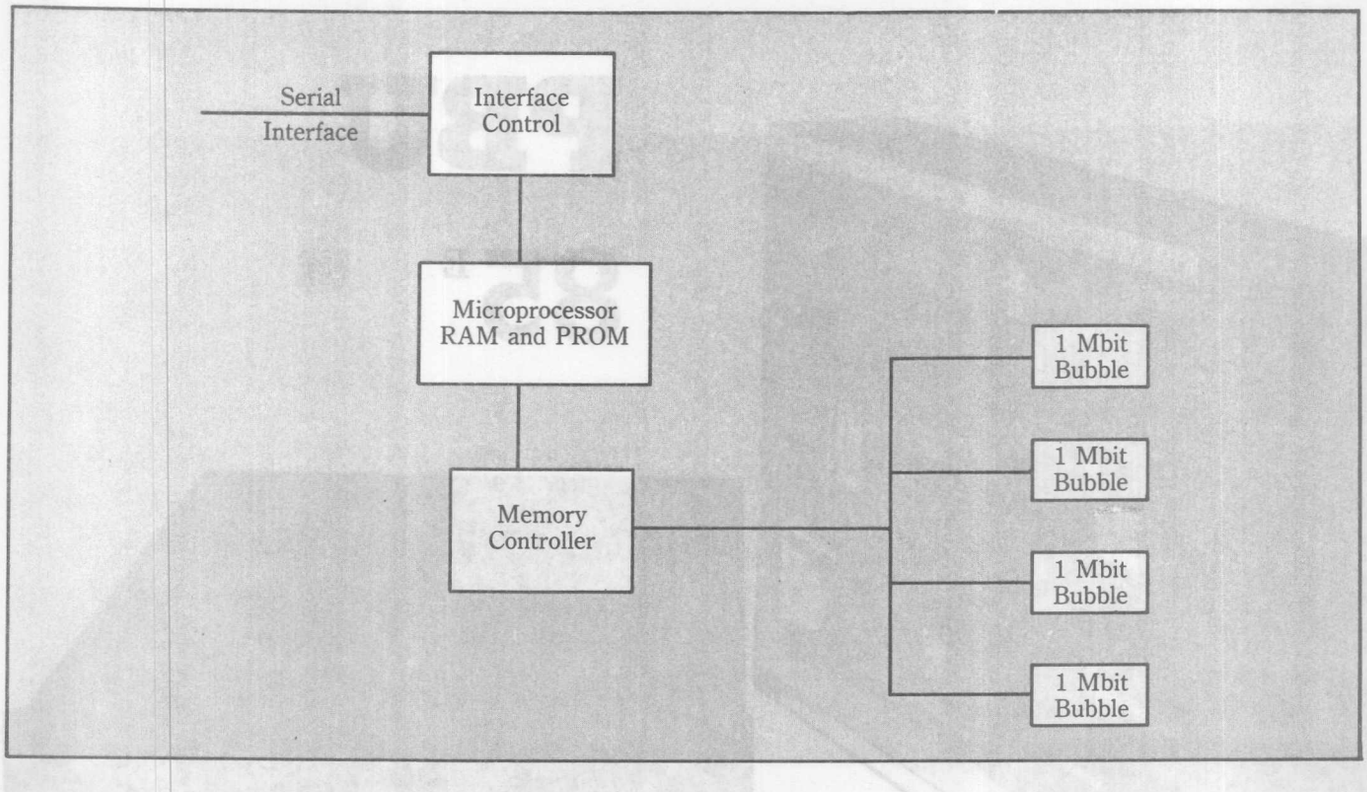
Reliability

PBU85E's capability to provide non-volatile storage over a wide range of temperatures makes it ideal for program loading or data storage applications where high integrity is important, particularly in a harsh environment.

Flexibility

Connection to the host is user selectable via RS 423 (RS 232C), or RS 422 serial interfaces. The data rate of the link is also selectable, between 75 baud and 38.4 kbaud.

The interface between PBU85E and the host machine is fully programmable. It can be used with most bit oriented protocols such as HDLC, SDLC and DEC TU58 tape RSP.



Specification

Onboard microprocessor handles data organisation, control formatting. Simple PROM based protocol for initialisation, position, read, write and test.

Capacity	512 kbytes or 256 kbytes. Random access, block addressable. Formatted in blocks of 512 or 128 bytes, software selectable.
Interface/buffer data rate	75 to 38.4 kbaud, selectable.
Bubble/buffer data rate	146 kbits/second.
Access time	Average 36 ms Minimum 26 ms Maximum 46 ms To the start of any block.
Test	Self test diagnostics
Error checking	16-bit add around check sum per 128 data bytes.
Interface	Serial, selectable to: RS 423 (RS 232C) or RS 422. TU58 compatible interface protocol, HDLC, SDLC options available.
Front panel indicators	Write protect, power on, access, power fail, status 0 to 15.
Data reliability	1 in 10^{12} bits read.
MTBF	Greater than 50,000 hours.
Power supply	+5V ±5% DC
Power requirements	Maximum standby current 0.5A Peak running current 3.0A Maximum inrush (switch on) current 7.0A
Dimensions	Width: 146 mm (5¾ inches) Height: 41 mm (1⅝ inches) Depth: 210 mm (8¼ inches)
Temperature range	Operating 0 to +55°C −20 to +70°C optional Storage −55 to +125°C Data retention −40 to +85°C
Relative humidity	Up to 95% without condensation
Thermal shock	±1°C/min throughout the operating temperature range.
Cooling	No cooling required at temperatures of less than 55°C, (70°C for wide temperature option)
Vibration	5 to 100 Hz with 0.5g acceleration
Mechanical shock	20g for 6 milliseconds (half sine)
Magnetic effect	20 Oersteds from any direction, dynamic or static
Maintenance requirement	Nil

PLESSEY
MICROSYSTEMS

Plessey Microsystems Limited
Water Lane, Towcester, Northants
United Kingdom NN12 7JN
Tel: Towcester (0327) 50312
Telex: 31628

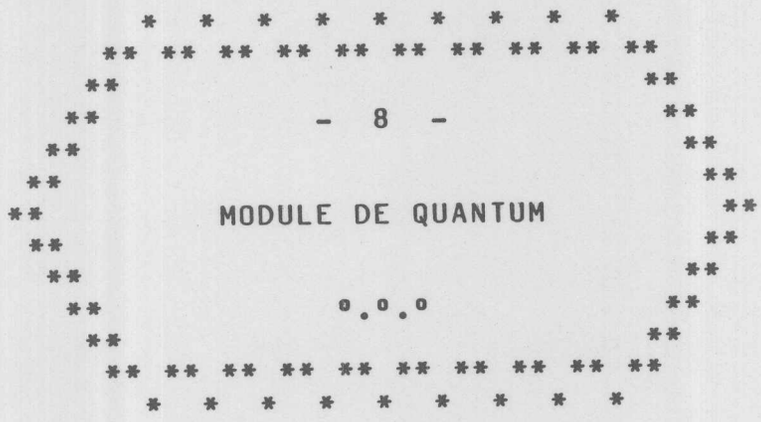
FRANCE
BP 74, 7-9 rue Denis Papin, 78194 Trappes Cédex
Tél: (3) 051.49.52 Télex: 696441

GERMANY
Bahnhofstraße 38, D-6090 Rüsselsheim
Tel: (0 61 42) 6 80 04 Telex: 17614293

USA
One Blue Hill Plaza, Pearl River, New York 10965
Tel: (914) 735 4661 Telex: 710 541 4661

451 Hungerford Drive, Rockville, Maryland 20850
Tel: (301) 279 2892 Telex: 710 828 9815

10900 East 183rd Street, Cerritos, California 90701


- 8 -
MODULE DE QUANTUM

- 8 -
MODULE DE QUANTUM

LES MODULES SBX OFFRENT L'AVANTAGE D'ETENDRE LES FONCTIONNALITES D'UNE CARTE SBC AU MOINDRE COUT.

iNTEL PROPOSE A SON CATALOGUE DES MODULES SBX ASSURANT LES EXTENSIONS SUIVANTES :

- E/S PARALLELE TOR EN TTL (SBX 251)
- E/S ANALOGIQUE (SBX 328, SBX 311)
- E/S SERIE RS 232 (SBX 351)
- E/S IEEE 488 (SBX 488)
- E/S SERIE HDLC (SBX 352)
- MEMOIRE A BULLE (SBX 251)
- CONTROLLEUR DE DISQUE SOUPLE (SBX 218)

AFIN DE PARFAIRE CE CATALOGUE, METROLOGIE S'EST ENTOUREE DU CONSTRUCTEUR QUANTUM QUI PROPOSE LES EXTENSIONS SUIVANTES :

- MODULE DE SAUVEGARDE DES REGISTRES PAR EEPROM (QBX-FOG-E)
- E/S SERIE DE 2 A 4 PORTS RS 232 (QBX-MSP)
- MODULE HORLOGE TEMPS REEL (QBX-TMR)
- MODULE "CHIEN DE GARDE" (QBX-FST)

CES MODULES PRESENTENT BEAUCOUP D'INTERET DANS LES APPLICATIONS INDUSTRIELLES.

EN EFFET :

- LES COUPURES SECTEURS PEUVENT INTERVENIR, IL CONVIENT PARFOIS DE PREVOIR UNE REPRISE DE LA COMMANDE DU PROCESS APRES AVOIR PRIS SOIN DE SAUVEGARDER LE CONTEXTE : LES MODULES QBX-FOG-C ET QBX-GOG-E PERMETTENT D'ATTEINDRE CET OBJECTIF.

- L'ENTRETIEN DU TEMPS EST IMPORTANT POUR DATER LES EVENEMENTS : LE MODULE QBX-TMR EST FAIT POUR CELA.

- EN CAS DE CONFIGURATION REDONDEE, LE PASSAGE DE L'UNITE PILOTE A L'UNITE ESCLAVE NECESSITTE UN PROTOCOLE : LE MODULE QBX-FST REMPLIT CETTE FONCTION.

PRODUCT SUMMARY
R & D ELECTRONICS (PTY) LTD

November 1983

QBX-PCC
PARALLEL I/O, CLOCK AND JULIAN CALENDAR

clock and Julian calendar
up to 13 buffered I/O lines
on board battery backup
full speed iSBX bus compatible
periodic interrupt capability

The QBX-PCC is an iSBX compatible module providing time and date information for a host computer. The information provided is seconds, minutes, hours, date, day of week, and month. Through software it is possible to include leap year adjustments and a facility for years. An on-board 40 bit shift register may store information in power fail situations.

In addition to the clock function up to 13 buffered I/O lines are provided. They consist of Port A and 5 bits of Port C of the Intel 8255A-5 peripheral device.
The remainder of this device is used to interface to the clock. It also allows interface to the SBX bus without wait state generation.

QBX-FOG/C
HARDWARE STACK AND NON VOLATILE RAM

non-volatile storage on iSBX* bus
organised as a hardware stack
256 bytes
any address may be accessed
stack counter value can be read
provision for off board standby battery
on board wait state generator

The QBX-FOG/C provides an ideal method of non-volatile storage for iSBC computers. The 256 bytes are configured as a hardware stack which allows added software flexibilty to a system in terms of parameter passing in subroutines. It also saves memory space since it occupies 16 I/O locations if configured as I/O mapped I/O.

The RAM may be accessed by programming an address on the stack counter and then "pushing" to write and "popping" to read. Each "push" or "pop" automatically updates the stack counter. Since the write or read functions occupy only one I/O location DMA techniques may be used to transfer blocs of data. An on board wait state generator allows full speed operation of the micro-processor.

The non-volatility of course means that various parameters may be stored during a power fail.

QBX-FOG/E
HARDWARE STACK AND NON-VOLATILE EEPROM

non-volatile storage on iSBX bus
organised as a hardware stack
64 (QBX-FOG/E-64) or 256 (QBX-FOG/E-256) bytes of RAM and "shadow" EEPROM
any address may be accessed
stack operation immeditely updates memory address
5V only EEPROM
on board wait state generator

The QBX-FOG/E provides an ideal method of non-volatile storage for iSBC computers. The memory is configured as a stack, but random access is possible to any stack location. The memory has two portions, RAM with a "shadow" EEPROM. There is a bit for bit correalation from the RAM to the EEPROM. Saving the RAM in the EEPROM overwrites the contents of the EEPROM with an identical image of the RAM. Recalling the EEPROM overwrites the contents of the RAM with an identical image of the contents of the EEPROM. The RAM may be used for normal storage operations as well. See the QBX-FOG/E users manual for possible applications.

The RAM may be accessed by programming an address on the stack counter and then "pushing" to write and "popping" to read. Each "push" or "pop" automatically updates the stack counter. When mounted in a connector configured as I/O mapped I/O the module occupies no memory space. An additional stack can also provide many advantages. Since the write or read functions occupy only one I/O location DMA techniques may be used to load or unload blocs of data. An on board wait state generator allows full speed operation of the microprocessor.

QBX-MSP
MULTI SERIAL PORT

2 (QBX-MSP/2) or 4 (QBX-MSP/4) asynchronous channels
uses Western Digital WD2123 devices
16 independent baud rates for each channel (50-19200 baud)
5-8 bits per character
1, 16, or 64 times clock rate sampling
line break detection and generation
odd and even parity generation and detection
double buffering of data
overrun and framing error detection
asynchronously compatible with industry standard 8251A
diagnostic local loop back mode
iSBX bus compatible (double sized board)
interrupt generation

on board wait state generation
RS232 voltage level drivers and receivers
QBN-MSP-CON available to provide connection to normal D-type connector

The QBX-MSP allows 2 or 4 asynchronous ports on a double sized SBX module. Each port has an individually programmed baud rate generator that allows transmission rates from 50 to 19200 baud. Once set up, the handling of the control and data ports is compatible with most software written for the 8251A serial communications controller in the asynchronous mode. A facilty exists for a local loop back test and interrupt gating is provided to generate an interrupt to the host processor. An on board wait state generator allows full speed operation of the microprocessor.

QBX-TMR
MULTI FUNCTION TIMING CONTROLLER

Sophisticated Timing Controller on single SBX* card.
5 independent 16 bit counters
High speed counting rates
Up/down and Binary/ BCD counting
Internal/external frequency source with tapped frequency scaler
Programmable frequency output
Time of day option
Alarm comparators on counters 1 and 2
Counters may be internally concatenated
Complex duty cycle outputs
One shot or continuous outputs
Programmable counter source selection
Programmable input and output polarities
Programmable gating
Retriggerable
Inputs overvoltage protected
Outputs buffered by Darlington transistors

Counting events and timing intervals are fudamental to many microcomputer projects. If the computer is required to execute several of these functions simultaneously in software, the time available for other requirements can become severely limted. By allowing these functions to be executed in hardware improves the situation considerably.

The QBX-TMR provides the Multibus user with these facilities by placing one of the most sophisticated timing controllers, the Am9513 on a single sized SBX card. This device has 5 independent 16 bit counters which may even be internally concatenated in stages to allow up to an 80 bit counter. For each counter there are 16 possible counter inputs selected by software. They include any of 5 source inputs, any of 5 gate inputs, or a selection from a the divided outputs of an external oscillator.

Up to 10 gates exist to allow modification of the counting patterns to generate monostable, FSK generation and many other functions. The input and output polarities are programmable.

The inputs are protected by a zener diode and series resistor whilst the outputs are buffered by Darlington drivers allowing medium voltage and current interface.
Interrupts are also catered for, and a latch exists to store transient pulses.

QBC-QBX
Quad SBX Module MULTIBUS Board

Uncommitted MULTIBUS board
Supports up to 4 SBX 8 bit modules
Memory Mapped or I/O mapped I/O
Address space to 20 bits
Programmable XACK response time
Configured as 8 bit MULTIBUS slave
IEEE P796 Bus Compliance: D8 M20 I16 VO L
Unoccupied modules do not occupy memory or I/O space
Does for the SBC card what the (E)PROM did for the ROM

The SBX bus introduced by Intel has allowed considerable flexibility in configuring MULTIBUS systems. Each module represents a specifc function and is used only when needed. This reduces the "overkill" of having to add a full size MULTIBUS card with unwanted functions. Only the newer MULTIBUS CPU cards have three connectors whilst others have less or even none. This means that if a function is required beyond the SBX capabilty of the host CPU board, a new card is needed, and so the advantage of the SBX modules is lost. The QBC-QBX addresses this problem.

The QBC-QBX is merely a host board for up to 4 SBX compatible boards (4 single, 3 single and 1 double, or 2 single and 2 double). It allows the board to be configured as a MULTIBUS slave, memory mapped or I/O mapped I/O. It will only support 8 bit SBX modules, but as 16 bit bus masters can handle 8 bit transfers this is not seen as a problem. As memory mapped I/O up to 20 bits of address may be decoded.

The QBC-QBX allows shared SBX facilities in a multiprocessor environment on the MULTIBUS. It also allows the retrofit of SBX modules to existing projects and even the use of stock (or cheaper) CPU boards that do not support the SBX bus on new projects.

QBX-FST
FAILSAFE TIMER

Dual monostable Failsafe Timer
Each monostable has separate address
Cold start / warm start detection
Visual indication of failsafe reset
Enable for external devices e.g. solenoids
AC low detection
Uncommitted single bit output
Outputs buffered by Darlington transistors

Almost any application involving microcomputers should include a failsafe timer for totally reliable operation. The cost associated with correcting erratic computer behaviour, be it due to unusual combination of input variables or simply radiated electronic noise, is normally very high. It requires a large number of man hours to wait for the fault that occurs only sporadically and unpredictably and ties up equipment that could be used more profitably elsewhere.

The principle of a failsafe timer is that periodically the microcomputer must signal the timer that it is functioning properly and delay consequential action for a further set period. If a failure occurs a reset pulse is generated and the processor is restarted. In this set of circumstances it may be necessary for the microcomputer to "know" whether it has just been switched on or if, in fact, a failure did occur. Expierience has also shown that even in a failed condition a system can still clock a failsafe timer.

The QBX-FST addresses all these problems within the Multibus environment. It provides two failsafe timers that are combined to generate a reset signal whenever either "times out". As a further level of security these two monstable timers are at different I/O locations and both require a 2 stage operation to restart the timer. In addition the QBX-FST also has the capabilty of providing information on whether the reset was due to switching on (cold start) or due to a failure reset (warm start). This information is also presented on Light Emitting Diodes for the system designer and the commissioning engineer can interpret the effects being observed.

On many systems where external devices such as solenoids and relays are controlled by the host computer these devices may behave in an unwanted manner until conditions in the computer (at switch on) have stabilised. On computer failure other unwanted conditions may arise. The QBX-FST has an output derived from the failsafe timers that will enable these devices only when they should be, and will turn them off on failure.

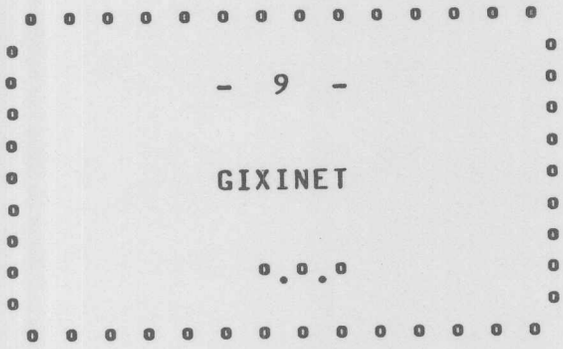
With the available board space a detector for AC low is implemented on the QBX-FST as well as a single bit buffered output for user convenience.

QBN-MSP-CON
RS232 D-TYPE CONNECTORS
FOR THE QBX-MSP

2 (QBN-MSP-CON/2) or 4 (QBN-MSP-CON/4) 25 way D-type subminiature female connectors to adapt the edge connector of a QBX-MSP to RS232 style connectors.
Height is low enough to mount on the back of the back panel of an Intel iSBC 655 System Chassis
50 cm (19.6 in) ribbon cable with mating card edge connector for QBX-MSP

DIL configurators to allow customisation of RS232 signal requirements

The QBN-MSP-CON is made to facilitate the connection of the standard RS232 style connector to the QBX-MSP. It is supplied in two versions to match the two versions of the QBX-MSP. The QBN-MSP-CON/2 is simply a depopulated (and therefore cheaper) version of the QBN-MSP-CON/4.
The QBN-MSP-CON consists of 2 or 4 25 way subminiature D-type female connectors with a matching number of configurator DIL sockets. The board is connected to the QBX-MSP via a 26 way ribbon cable and edge connector.
For normal applications opposing pins of the configurator are connected together. However, by varying the connections on the configurator, the QBN-MSP-CON may be customised to a particular requirement.


- 9 -
GIXINET

METROLOGIE EST EN COURS DE SIGNATURE D'UN ACCORD AVEC GIXI RELATIF A LA COMMERCIALISATION DES PRODUITS GIXINET. IL S'AGIT DE MATERIELS ET DE LOGICIELS PERMETTANT DE REALISER UN RESEAU LOCAL POUR MILIEU INDUSTRIEL OU DE BUREAUTIQUE. CE RESEAU A DEJA ETE RETENU PAR UN GRAND NOMBRE DE SOCIETES. C'EST PROBABLEMENT CELUI QUI REMPORTE LE PLUS DE SUCCES EN FRANCE. L'INTERET DE LA SOLUTION GIXI EST DONC D'EXISTER ET D'ETRE OUVERTE VERS L'AVENIR PUISQUE DES ETUDES SUR LES PRODUITS QUI SERONT INDUSTRIALISES DANS LES 10 ANS A VENIR SONT DEJA EN COURS.

I - HISTORIQUE

En 1974, une petite SSCI (CERIA) déposait en France un brevet (dont les co-inventeurs étaient Jean Suchard et Jean-Michel COUR l'auteur de cet article) pour un "système de transmission entre plusieurs stations" et sa marque "Table Ronde". Les brevets étaient ensuite déposés pour l'Amérique de Nord, l'Europe et le Japon. A cette époque, le terme générique de réseau local (local area network) n'était pas encore employé, mais "les idées étaient dans l'air". Par exemple, IBM avait déjà décrit la boucle SLDC ; l'année suivante, Xerox (R. Metcalfe) brevetait le futur Ethernet.
En 1976, une première maquette démontra la faisabilité du procédé "Table Ronde", avec une vitesse de transmission de 10 K bits par seconde qui nous semble aujourd'hui bien modeste. En 1978/1979, c'est dans le cadre du goupe CISI et plus particulièrement dans GIXI, sa filiale d'ingénierie de systèmes, que fut développé le premier modèle "commercial" du réseau : TR 78. Initialement demandés par l'Iria (devenue Inria) pour le démarrage du projet pilote Kayak, les communicateurs TR 78 (unités d'accès en réseau) furent produits en une vingtaine d'exemplaires, livrés principalement à des équipes de recherche.
Aujourd'hui, avec les modèles TR 81, le système de réseau local Gixinet se veut compétitif avec les systèmes d'origine américaine (Net 1, Z.Net et autres), et particulièrement dans le rapport service/prix.

II - LE PROCEDE TABLE-RONDE

Le système "Table Ronde" (abrégé en TR) est un procédé de réseau local à commutation de paquets. C'est à dire que son but est d'acheminer, entre différentes stations, des données groupées en blocs.
Contrairement à la plupart des autres systèmes, TR se veut largement indépendant de la topologie et de la nature des supports de commnication. Les supports (média) doivent simplement assurer une fonction de diffusion. Autrement dit, quand l'émetteur d'une des stations transmet, toutes les autres stations doivent recevoir le même message simultanément (au délai de propagation près).

A titre de premier exemple (fig.1), les stations peuvent être en dérivation sur un bus-série passif : coaxial, paire torsadée terminée, etc. Selon un second exemple (fig. 2), les stations sont reliées en point à point à un diffuseur actif. Cette seconde configuration contient d'ailleurs en germe la possibilité de changement de support.
Comme on le verra plus loin, le procédé est déterministe (pas de "collision" à détecter). Il s'ensuit que toute combinaison de supports est effectivement admissible sans précautions particulières.

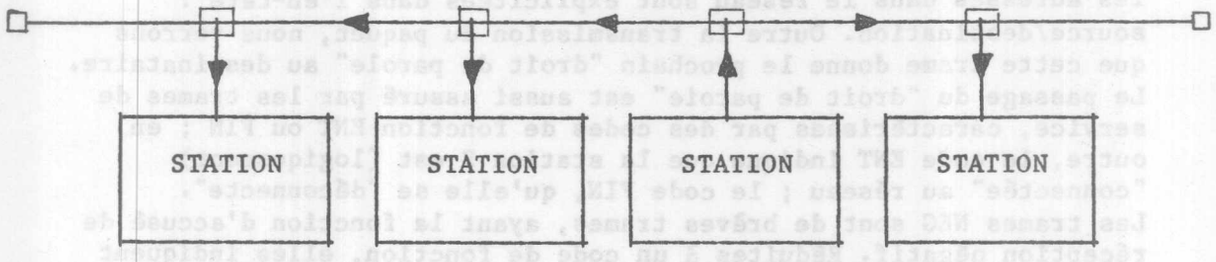


Fig. 1 : Diffusion sur bus (passif)

Exemple multipoint au standard V 11 (X27)

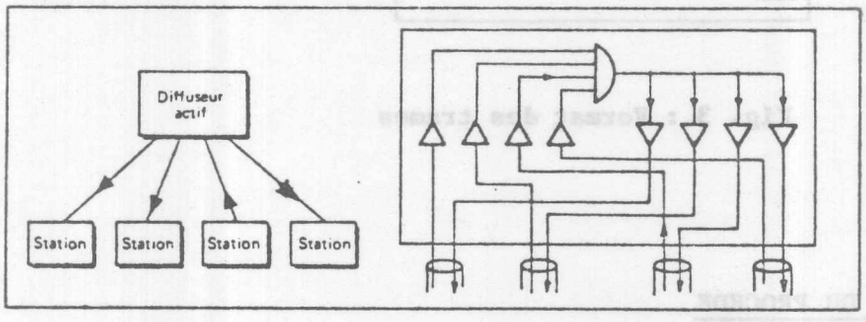


Fig. 2 : Diffusion en étoile (à gauche) et détail du diffuseur actif : les liaisons sont duplex, point à point

FORMAT DES TRAMES

Les trames véhiculées par les média entre les stations sont de trois formats (fig. 3). La trame d'information, caractérisée par le code-fonction DON, porte les "données" du niveau supérieur (paquet) ; les adresses dans le réseau sont explicitées dans l'en-tête : source/destination. Outre la transmission du paquet, nous verrons que cette trame donne le prochain "droit de parole" au destinataire. Le passage du "droit de paroie" est aussi assuré par les trames de service, caractérisées par des codes de fonction ENT ou FIN ; en outre, le code ENT indique que la station S est (logiquement) "connectée" au réseau ; le code FIN, qu'elle se "déconnecte". Les trames NEG sont de brèves trames, ayant la fonction d'accusé de réception négatif. Réduites à un code de fonction, elles indiquent un rejet pour erreur dans la réception d'une trame d'information (NEG2), ou pour cause de manque de ressource (NEG1).

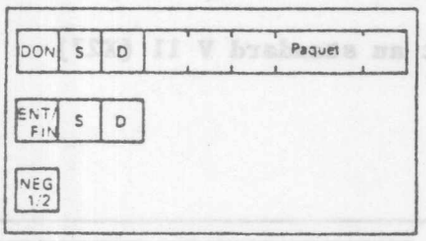


Fig. 3 : Format des trames

PRINCIPE DU PROCEDE

Le procédé repose sur la circulation déterministe d'un droit de parole, entre des stations ayant chacune une adresse propre. A un instant donné, ce droit appartient à une station et à une seule. En des circonstances normales, cette station transmet ce droit à une autre station (dans un délai maximal déterminé), laquelle le transmet ensuite à une autre station, et ainsi de suite.

Il existe deux façons pour une station d'utiliser son droit de parole :

a) Un paquet est en attente de transmission ; c'est alors une trame de type DON qui est transmise (Fig. 4), avec l'adresse de la station destinataire (quelconque dans l'ensemble des stations interconnectées).

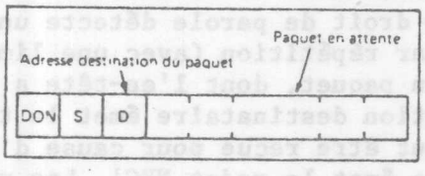


Fig. 4 : Emission d'une trame de type DON pour transmission d'un paquet en attente

b) Pas de paquet en attente ; dans des circonstances normales, c'est une trame de type ENT qui est constituée (fig. 5) ; l'adresse destination est prise dans une table : c'est (dans le cas le plus simple) l'adresse de la station suivante dans cette table.
Dans les deux cas (a) et (b), c'est la station destinataire qui prend le droit de parole, dès la fin de réception de la trame DON ou ENT.
En dehors de circonstances exceptionnelles (erreur de transmission, panne subite d'une station, etc...), le trafic consiste donc en une suite de trames qui "font circuler" le droit de parole. Certaines, en outre, véhiculent des paquets.

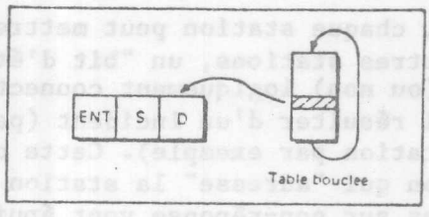


Fig. 5 : Emission d'une trame de type ENT

DETECTION ET REPRISE DES INCIDENTS

Classiquement, les trames véhiculent des bits redondants, pour la détection d'erreurs de transmission éventuelles. D'autre part, les trames doivent (par principe) se succéder à intervales rapprochés, c'est-à-dire dans un délai inférieur à un certain maximum ; si ce délai maximal expire ("time-out" classique), il y a "non réponse". Ce n'est pas indispensable, mais l'on préfère avoir des contrôles d'erreur distincts sur les en-têtes (code fonction, adresses source et destination), et sur les paquets.
Si une erreur affecte l'en-tête d'une trame reçue, cette trame est ignorée : c'est le cas notamment de la station destinataire : la station qui détenait le droit de parole détecte une non-réponse. La reprise a alors lieu, par répétition (avec une limite par comptage).
Si une erreur affecte un paquet, dont l'en-tête a néanmoins été correctement reçue, la station destinataire émet la trame de rejet NEG2. Si la trame ne peut être reçue pour cause d'engorgement (tampon pas libre), elle émet le rejet NEG1. Les reprises se font par répétition, avec (là encore) des limites par comptage.
Ces différentes modalités de détection et de correction des incidents sont classiques dans les procédures usuelles (BSC, HDLC par exemple). Elles sont ici exécutées "à la vitesse des média" ce qui n'est pas le cas d'autres procédés de réseau local (CSMA-CD par exemple).

NOTION D'ACTIVITE DES STATIONS

Par définition, une station d'un réseau donné est active si elle "prend" et "utilise" le droit d'émettre. Sinon elle est inactive, ce qui signifie que tout se passe comme si elle était physiquement déconnectée. Un exemple type sera une station solidaire d'un terminal, active si le terminal est "en ligne", inactive s'il est coupé.
Une station active qui va devenir inactive, c'est-à-dire se déconnecter "logiquement", transmet dès que possible la trame FIN :

Cette trame a les fonctions de la trame ENT (passage de droit) et de plus indique à toutes les autres la désactivation de la source "S".
Grâce à cette convention, chaque station peut mettre constamment à jour, pour chacune des autres stations, un "bit d'état" ; ces bits indiquent si elles sont (ou non) logiquement connectées.
La déconnexion peut aussi résulter d'un incident (panne d'alimentation sur une station par exemple). Cette circonstance est reconnue par toute station qui "adresse" la station défaillante : les mécanismes de reprises sur non-réponse vont épuiser leur compte maximal. En ce cas, le bit d'état correspondant est mis à l'état "inactif".
Outre l'intérêt que présente cet indicateur pour les stations (qui peuvent en quelque sorte savoir si leur "niveau trame" est établi ou non avec chacune des autres stations), le procédé exploite la table des bits d'activité pour optimiser le trafic.

OPTIMISATION ET DEMARRAGE

Dans le processus de passage du droit à la parole, les stations envoient en fait l'essentiel des trames ENT à la première station suivante active (fig. 6). Ainsi sont évitées des trames ENT inutiles, des délais de non-réponse, etc.
On trouve là une propriété souhaitée dans beaucoup de systèmes de réseau local, que les stations "qui en ont besoin" puissent bénéficier de "toute la bande passante" disponible à un moment donné.
Cependant, chaque station fera de temps en temps une entorse à la règle, c'est-à-dire qu'elle enverra une trame ENT à la station suivante même si elle est marquée inactive.
Cette opération, appelée réveil, permet aux stations qui viennent d'être activées de recevoir le droit de parole une première fois.
Dans les implémentations courantes, "de temps en temps" est interprété comme "1 fois sur N", où N est le nombre d'opérations de passage du droit à la parole.
Naturellement, l'opération de réveil ne fait pas, en elle-même, l'objet de reprises (une seule tentative à la fois). Si elle échoue, la station responsable continue avec l'adresse de la station suivante dans la table.
Le démarrage du système n'est qu'un cas particulier de réveil. La première station activée reconnait qu'elle est "seule" en "écoutant le réseau" et en constatant (sur délai) qu'il n'y circule aucune trame. Elle s'octroie alors le droit de parole... et ne l'abandonne que lorsqu'un réveil (au moins) réussit ; le processus déjà décrit fait que la première sttion activée effectue (littéralement) des "tours de table" jusqu'à obtenir au moins un enchainement par une autre station nouvellement activée.

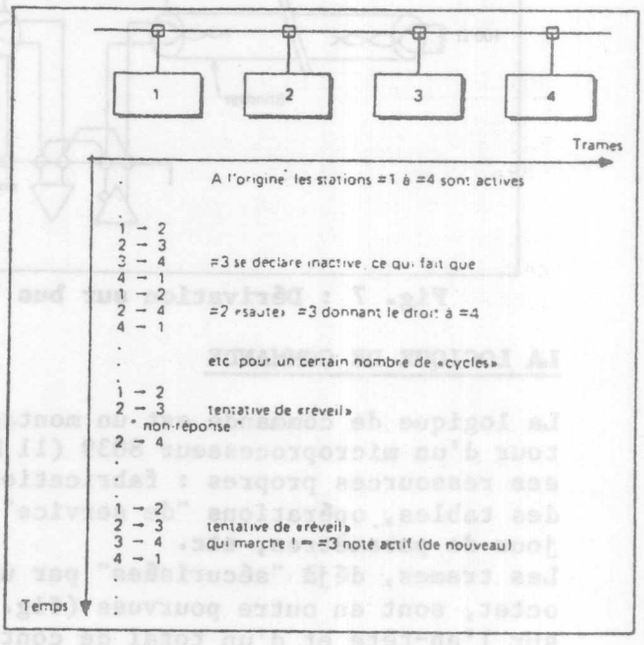


Fig. 6 : "Tranche de vie" d'un réseau TR (sans transmission de paquet pour alléger la figure) les notations a -- b correspondent à l'envoi de la trame ENT a dest b.

III - REALISATION DU "COMMUNICATEUR"

Les principes ci-dessus décrits sont très généraux, et peuvent être repris pour toute une variété de réalisations possibles.
Pour donner corps au procédé, on peut notamment réaliser une "boîte noire" ou communicateur qui :

- contient les interfaces pour accès aux média (interface de ligne) ;
- comporte les circuits logiques pour réaliser le "protocole" Table Ronde ;
- présente aux usagers (ordinateurs) une interface qui les "isole", dans le temps, des détails de ce protocole.

COMMUNICATEUR TR 81 : INTERFACE DE LIGNE

L'interface de ligne est connectée sur un très banal circuit UART, transmetteur asynchrone exploité sur la base de 125 K bits par seconde. La transmission se fait en série par octets, avec Start, Stop et Parité. Le choix de ce mode de transmission est essentiellement motivé par des soucis de simplicité (notamment, en termes de synchronisation).
Les circuits transmetteurs et récepteurs proprement dits sont au standard V11 (alias RS 422). Les signaux différentiels ont des avantages bien connus (rejection de bruit de mode commun notamment). En outre, le transmetteur a une entrée de validation qui permet de le placer en dérivation sur bus-série. Ainsi, en bas de gamme, on peut constituer le médium aussi simplement que décrit en figure 7 : le bus-série est passif.

Les stations sont connectées via des... joncteurs téléphoniques usuels.
Les limites paramétriques admises sont d'environ vingt stations connectées par un câble d'environ 500 m du type paire torsadée, blindée, terminée sur une résistance de 100 Ω . Au delà, ou pour des cas spéciaux (spécifications d'isolement, etc.), et ou pour changement de support (câble coaxial ou fibre optique par exemple), on réalise des montages avec répéteurs/diffuseurs actifs. La vitesse de 125 K bits par seconde a, entre autres avantages, celui d'autoriser l'emploi d'un support bon marché (paire torsadée blindée) et/ou de rendre peu critiques les différentes connexions physiques, changement de supports, etc.

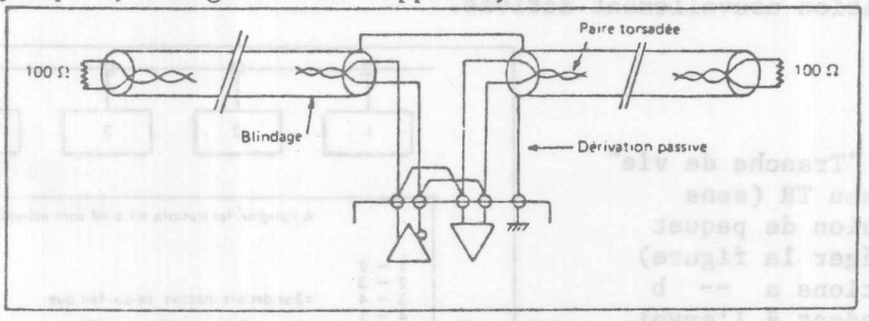


Fig. 7 : Dérivation sur bus "deux fils"

LA LOGIQUE DE COMMANDE

La logique de commande est un montage très classique, réalisé autour d'un microprocesseur 8039 (11 MHz) qui prend en charge, avec ses ressources propres : fabrication et analyse des trames, gestion des tables, opérations "de service" telles que diagnostics, mise à jour de paramètres, etc.
Les trames, déjà "sécurisées" par un bit de parité pour chaque octet, sont en outre pourvues (fig. 8) d'un total de contrôle (VRC) sur l'en-tête et d'un total de contrôle sur le paquet.
En outre, des contrôles de vraisemblance sont effectués partout où cela est possible : code-fonction, adresses, longueur effective du paquet = longueur indiquée.

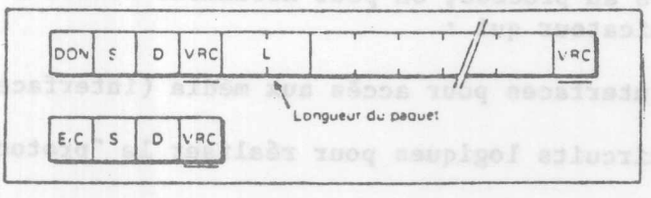


Fig. 8 : Structure détaillée des trames

Le modèle TR 81 de base supporte jusqu'à 64 adresses ; soit 64 stations au plus. Cela dit, on peut prémarquer (en Prom) des configurations plus restreintes. En outre, une adresse de diffusion est offerte. Elle permet d'envoyer le même paquet à toutes les autres stations en une seule fois ; contrepartie, les services de reprises d'erreur sont inhibés (c'est-à-dire que le bon acheminement n'est pas garanti).
Le délai maximal du protocole est très petit par rapport aux procédures usuelles. A titre d'exemple, la non-réponse est détectée sur une temporisation de l'ordre de 1 ms. Dans le cas du plus grand réseau (64 stations) dont une seule "scrute" toutes les autres, il s'ensuit que le cycle de scrutation est inférieur à 100 ms.
Hors opérations de réveil, et quand aucun paquet n'est transmis, le droit de parole circule à raison de 700 us environ par station. Un système (hypothétique) à 64 stations toutes actives serait à l'état de "repos" complètement "scruté" en 50 ms environ.

LA LOGIQUE DE DIALOGUE

La logique de dialogue a été conçue avec les objectifs suivants : hautes performances (débit "local" élevé) ; complet asynchronisme par rapport au protocole de ligne ; interfaçage matériel aussi simple que possible vis-à-vis des mini et micro-ordinateurs.
Les deux éléments d'échange des paquets sont des mémoires-tampon à "double accès dans un seul sens", chacune constituée par une Ram (de 1 K octet pour TR 81). L'adressage est réalisé par un compteur, qui peut être remis à zéro par les deux côtés : "fournisseur" et "consommateur".
L'écriture d'un octet, d'une part, et sa lecture, d'autre part, font toutes deux progresser le compteur de 1.
Ainsi, les tampons sont accessibles "en série par octet" et aussi "recyclables" par remise à zéro du compteur d'adresse.

LE MODE D'EMPLOI

Sans décrire toutes les fonctions comme dans le manuel de référence, on peut suggérer les fonctions essentielles par le "scénario" suivant :

- l'ordinateur "usager" effectue la commande 6 : ce qui restaure (Reset) le communicateur. Celui-ci est alors (ou est encore) déconnecté du réseau, c'est-à-dire inactif ;

- par la commande 4, l'usager demande la connexion logique du communicateur, qui devient actif pour les autres stations ;

- la réception de paquets est très simple : dès que le mot d'état indique une fin de réception, l'usager remet à zéro (commande 0) le pointeur de la MTR, et recopie séquentiellement le paquet dans la mémoire de travail. La MTR est rendue au communicateur par la commande 1 ;

- l'émission de paquets est un processus symétrique : l'usager remet à zéro (commande 2) le pointeur de la MTE, et garnit cette mémoire séquentiellemen. Par la commande 3, l'émission est demandée. On attend alors que le mot d'état indique la fin d'émission ; le compte-rendu dira si la transmission s'est bien passée, ou a éventuellement échoué (épuisement des reprises) ;

- la déconnexion est demandée par la commande 5.

Dans le découpage en "couches" des réseaux, qui commence à entrer dans les moeurs, et spécialement sous l'impulsion de l'ISO avec le modèle "OSI", le niveau de service atteint par le communicateur est soit celui de la couche 2 (DATA LINK) si des passerelles vers les réseaux publics sont utilisées, soit celui de la couche 4 si les communications restent internes au réseau local. Notamment, parce que sont assurées les fonctions : d'adressage, de contrôle et reprise des erreurs de transmission, de premier niveau de contrôle d'engorgement (flow control).
On peut remarquer que, pour la plupart des autres systèmes de réseau local, l'unité de couplage au réseau (grossièrement, l'équivalent de notre "communicateur") ne va pas aussi loin dans les services rendus. Il est alors demandé au logiciel de l'ordinateur "hôte" (usager) d'assurer certaines fonctions de niveau 2, comme les reprises ... ce qui affecte d'autant les performances "à service égal". Par exemple, Ethernet (et les systèmes CSMA-CD en général) ne corrigent pas les erreurs de transmission ou les pertes par engorgement.
Ces différences de niveau de service ne sont pas des subtilités académiques. Elles ont un impact direct sur les coûts et/ou les performances et sur le service. Les conséquences d'une amélioration de celui-ci sont évidentes : moins de logiciel à développer dans les ordinateurs interconnectés ; performances à sécurité égale obtenues avec une vitesse moindre, donc une infrastructure plus "rustique" (moins chère in fine) et adaptation presque immédiate aux réseaux de type X25 (le niveau de service étant pratiquement le même que celui des procédures HDLC).

IV - LES PRODUITS GIXINET

Les communicateurs TR 81 se présentent sous forme de cartes au format SBC :

- TR 81A0 : communicateur au format SBC sur lequel un mini ou micro ordinateur se connecte par une interface parallèle.
  Dans un système SBC, la liaison avec une carte unité centrale se fait par un module iSBX (TR 81 A2). Les handlers pour RMX86 sont disponibles. Pour d'autres systèmes, le communicateur est fourni dans un coffret avec alimentation (TR 81 A1). La liaison se fait généralement en utilisant un GPI (SOLAR, MITRA, APPLE ...).

- TR 81 M : carte au format SBC comprenant un communicateur, fonctionnellement identique à TR 81 A0, et un processeur de service 8085 pouvant avoir jusqu'à 32 Ko de mémoire morte et 8 Ko de mémoire vive.
  Le raccordement des équipements sur TR 81 M se fait par un module iSBX. En particulier le module TR 81 M1 fourni une interface asynchrone simplifiée. Les communicateurs TR 81 M sont fournis sans logiciel sur le processeur de service.

Jean-Michel COUR
Olivier GIBERGUES

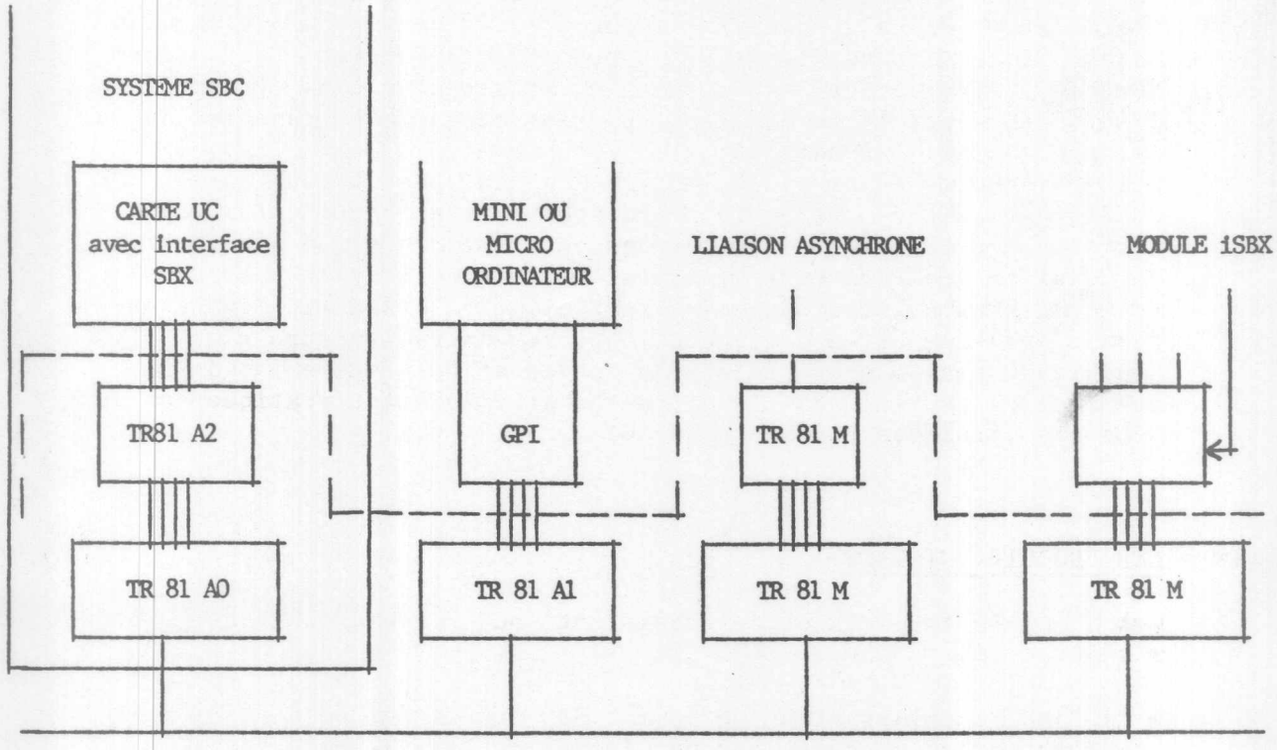


Fig. 9 : Exemples de raccordements

— — —limites de fourniture GIXINET

Fig. 9 : Exemples de raccordements

— — —Limites de fourniture GIXINET